Master's Thesis

Title

Studies on Congestion Control Mechanisms in the Internet – AIMD-based Window Flow Control Mechanism and Active Queue Management Mechanism –

Supervisor

Prof. Masayuki Murata

Author

Motohisa Kisimoto

February 12th, 2003

Department of Informatics and Mathematical Science

Graduate School of Engineering Science

Osaka University

Master's Thesis

Studies on Congestion Control Mechanisms in the Internet

– AIMD-based Window Flow Control Mechanism and Active Queue Management Mechanism –

Motohisa Kisimoto

Abstract

AQM (Active Queue Management) mechanisms support an end-to-end congestion control mechanism of TCP (Transmission Control Protocol). Researchers have claimed several advantages of AQM mechanisms over a conventional Drop-Tail gateway such as the small average queue length (i.e., the number of packets in the buffer). Several AQM mechanisms have been recently proposed and studied by many researchers. One of promising AQM mechanisms is the RED (Random Early Detection) gateway, which randomly discards arriving packets. Although its steady state performance has been fully investigated, its transient behavior has not been well understood. RED randomly drops the arriving packet with a probability proportional to its average queue length. However, it is unclear whether the packet marking function of RED is optimal or not. In this paper, we first analyze the transient behavior of the RED gateway for various types of TCP connections variations. We use a control theoretic approach by utilizing the transfer function, which describes the relation between the input and the output in frequency domain. By presenting several numerical and simulation results, we discuss how control parameters of the RED gateway affect its transfer behavior. We also investigate what type of packet marking function, which determines the packet dropping probability from the average queue length, is suitable from the viewpoint of steady state performance and transient behavior. By presenting several numerical examples, we investigate advantages and disadvantages of three packet marking functions: linear, concave, and convex. We show through numerical examples that although the average queue

length in steady state becomes larger, use of the concave function makes the transient behavior of RED and the robustness against network status changes better.

Keywords

Internet, TCP (Transmission Control Protocol), RED (Random Early Detection), Active Queue Management Mechanism, Transient Behavior, Packet Marking Function, Steady State Performance, Robustness

Contents

1 Introduction **8**

2 RED (Random Early Detection) **13**

3 Transient Behavior Analysis of RED using a Control Theoretic Approach **15**

3.1 Analytic Model 15

3.2 Average State Transition Equations 16

3.2.1 Derivation of State Transition Equations 17

3.2.2 Derivation of Average State Transition Equations 18

3.3 Transient Behavior Analysis 20

3.4 Numerical Examples and Discussions 25

3.4.1 Performance Measures for Transient Behavior 26

3.4.2 Overshoot, Rise Time, and Settling Time 27

3.4.3 Stability Analysis 32

3.4.4 Transient Behavior Analysis using Transfer Function 33

4 Packet Marking Function of Active Queue Management Mechanism: Should It Be Linear, Concave, or Convex? **38**

4.1 Analysis 38

4.2 Numerical Examples and Discussions 44

4.2.1 Case of RED (Random Early Detection) 44

4.2.2 Case of Adaptive RED 50

5 Conclusion **55**

Acknowledgements **57**

References **58**

List of Figures

1 Packet marking function of RED — calculation of packet marking probability p_b from average queue length $\overline{q}$. 14

2 Analytic model. 15

3 Relationship between slot and sequence. 19

4 Cases C1 and C2 where all TCP connections in congestion avoidance phase. . . . 22

5 Cases C3 and C4 where a part of TCP connections in slow start phase. 23

6 Performance measures for transient behavior (overshoot, rise time, and settling time). 27

7 Performance measures for transient behavior (the number of previous TCP connections N = 1–7). 28

8 Performance measures for transient behavior (the processing speed of the RED gateway B = 1–10 [packet/ms]) 29

9 Performance measures for transient behavior (the propagation delay of the TCP connection τ = 1–5 [ms]) 30

10 Performance measures for transient behavior (the maximum packet marking probability max_p =0.025–0.15) 31

11 Performance measures for transient behavior (the number of resumed TCP connections ΔN = 1–10) 32

12 The maximum modulus of poles in the min_{th}–max_{th} plane. 33

13 The maximum modulus of poles in the B–τ plane. 34

14 Impulse response and gain characteristic of $U(z) \times G(z)$ for $\Delta N_1 = 1$. 35

15 Impulse response and gain characteristic of $U(z) \times G(z)$ for $(\Delta N_1, \Delta N_2, \Delta N_3) = (1, 1, 1)$. 36

16 Impulse response and gain characteristic of $U(z) \times G(z)$ for $(\Delta N_1, \Delta N_2, \Delta N_3) = (1, 2, 1)$. 37
17 Comparison between analytic and simulation results. 37
18 Evolution of TCP window size in congestion avoidance phase. 40
19 Queue occupancy in the x-x^* plane. 42
20 RED queue occupancy in the x–x^* plane with $\mathcal{F}_\phi$ (linear) for ϕ = 0.5, 1.0, and 1.5. 47
21 RED queue occupancy in the x–x^* plane with $\mathcal{G}_\phi$ (concave) for ϕ =0.5, 1.0, and 1.5. 48
22 RED queue occupancy in the x–x^* plane with $\mathcal{H}_\phi$ (convex) for ϕ = 0.5, 1, and 1.5. 49
23 Relation between the number of active TCP connections N and RED queue occupancy with $\mathcal{F}_\phi$ (linear) for ϕ =1.0. 50
24 Relation between the number of active TCP connections N and RED queue occupancy with $\mathcal{G}_\phi$ (concave) for ϕ = 1.0. 51
25 Relation between the number of TCP connections N and RED queue occupancy with $\mathcal{H}_\phi$ (convex) for ϕ = 1.0. 52
26 Adaptive RED queue occupancy in the x–x^* plane for $N = 5$ and $max_p = 0.02$. 53
27 Adaptive RED queue occupancy in the x–x^* plane for $N = 10$ and $max_p = 0.1$. 54
28 Adaptive RED queue occupancy in the x–x^* plane for $N = 20$ and $max_p = 0.4$. 54

List of Tables

1 Definitions of symbols. 16

2 Parameters used in numerical examples. 46

1 Introduction

In the last few years, an AQM (Active Queue Management) mechanism, which supports an end-to-end congestion control mechanism of TCP, has been studied by many researchers [1]. For instance, a RED (Random Early Detection) is a representative AQM mechanism, which randomly drops an arriving packet at the gateway for improving the performance of TCP traffic [2]. 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, (2) the performance degradation caused by a global synchronization problem found in the Drop-Tail gateway is avoided, and (3) the RED gateway improves the fairness among TCP connections. Although the effectiveness of the RED gateway is fully dependent on a choice of its four control parameters, it is difficult to configure them [2, 10, 15, 3].

A number of studies on the RED gateway have been extensively performed by many researchers. Most of those studies (e.g., [2, 3]) have used a simulation technique for clearly revealing characteristics of the RED gateway in various network configurations and for investigating how control parameters of the RED gateway affect its efficiency. There have been, however, only a few analytical studies on the RED gateway. In [4-8], the performance evaluations of the RED gateway in the steady state have been performed. The authors of [2] have proposed a recommended set of control parameters, which is an empirical guideline by simulation experiments. The authors of [5, 8] have proposed another guideline, which is based on their analytic results.

Although there have been a great number of researches on the RED gateway, most of them simply focus on its *steady state behavior*. There have been very few researches on the *transient behavior* of the RED gateway. Stability and transient behavior of the RED gateway in the steady state have been analyzed in [6-9] 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 a real network, the number of TCP connections changes fre-

quently. 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.

As explained above, RED has a problem that its effectiveness is greatly dependent on a setting of four control parameters such as min_{th}, max_{th}, max_p, and q_w, and effectiveness of RED has been denied by many researchers [3, 10]. Another problem of RED is that the average queue length of RED is dependent on the number of active TCP connections; i.e., the optimal setting of RED control parameters changes according to the number of TCP connections. For solving such problems of RED, various researches have so far been devoted [11-15]. For example, SRED (Stabilized RED) is proposed in [14]. SRED estimates the number of active TCP connections at the router, and controls its average queue length regardless of the number of active TCP connections. In SRED, the packet loss probability is determined based on the estimated number of TCP connections. Moreover, Adaptive RED is proposed in [15]. Adaptive RED dynamically increases or decreases the maximum packet loss probability max_p, which is one of RED control parameters, according to the average queue length. When the average queue length is smaller than min_{th}, Adaptive RED decreases max_p by $(1/\alpha - 1)\, max_p$. On the other hand, when the average queue length is larger than max_{th}, max_p is increased by $(1 - \beta)\, max_p$.

These approaches for solving several problems of RED are only ad hoc approaches. This is because those problems of RED originate from the fact that the algorithm of RED is designed by an ad hoc approach. For instance, RED randomly discards an arriving packet with a probability that is proportional to the average queue length. Considering purposes of AQM mechanisms, it seems plausible to increase the packet loss probability when the average queue length is large, and to decrease when the average queue length is small. However, little investigation has been done for clarifying "whether the packet loss probability should be proportional to the average queue length or not". Originally, RED randomly discards an arriving packet with a probability,

which is determined by a linear function to the average queue length. More specifically, when the difference between the average queue length and min_{th} increases by α, RED increases its packet loss probability by α. However, it is known that the throughput of TCP is inversely proportion to $\sqrt{p}$, which is the packet loss probability in the network [16]. Also known is that, provided that the bottleneck router can be modeled by a single M/M/1 queue, the average queue length of the bottleneck router is given by $\rho/(1-\rho)$, which is the utilization of the network [17]. Hence, considering that the primary purpose of AQM mechanisms is to control the average queue length, it is expected that the packet loss probability of RED should not be changed linearly to the average queue length. Of course, rather than using a simple queuing model such as M/M/1, it is necessary to model the interaction between the TCP congestion control mechanism and the network.

In the literature [8, 4, 18], there have recently been analytical studies on AQM mechanisms, which model both the TCP congestion control mechanism and the AQM mechanism. For instance, in [8], TCP and RED are first modeled as independent discrete systems. By combining these two systems, the entire network is modeled as a single feedback system between TCP and RED. By applying control theory, the authors of this paper analyze stability, steady state behavior, and transient behavior of RED. Moreover, in reference [4], TCP and RED are modeled as independent continuous systems, and stability and steady state behavior of RED are analyzed. In particular, in [18], a new AQM mechanism is proposed by applying classical control theory to the mathematical model of TCP. In [18], the PI (Proportional Integral) controller in classical control theory, rather than an ad hoc approach, is proposed as the AQM mechanism. However, the PI controller is a simple linear controller that uses the difference between the average queue length and the target queue length. Still, there is little investigation on how the packet loss probability should be determined from the average queue length; i.e., the function determining the packet loss probability should be a linear function to the average queue length?

In the first part of this thesis, we analyze the transient behavior of the RED gateway by extending the analytic results obtained in [8]. 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. We analyze the transient behavior of the RED gateway for various types of TCP connections variations. We use a control theoretic approach by utilizing the transfer function, which describes the relation between the input and the output in frequency domain. We also validate our approximate analysis on the transient behavior of the RED gateway by comparing analytic results with simulation ones. Showing numerical results, we reveal how control parameters of the RED gateway affect its transient behavior. However, it has not been cleared whether the packet marking probability should be proportional to the average queue length or not.

In the second part of this thesis, using an analytic approach, we therefore investigate how the function determining the packet loss probability affects the RED performance for evaluating its impact on steady state behavior and transient behavior of RED. Specifically, we utilize results of the TCP steady state analysis in [16] and the RED steady state analysis in [8], and show how the packet loss probability function of RED should be determined. In numerical examples, we consider three classes of functions as the packet loss probability function — linear, concave, and convex — and show how the RED performance is affected by a choice of the packet loss probability function. We show that, when the packet loss probability function is concave, the transient behavior and the robustness of RED are improved compared with the case of a linear function, which is adopted by the original RED.

In addition, by applying the analysis method proposed in this thesis to Adaptive RED, we discuss steady state behavior and transient behavior of Adaptive RED. The algorithm of Adaptive RED is essentially the same with that of RED except that the maximum packet loss probability max_p is dynamically changed according to the network status for realizing robustness [19]. Our analytic result shows that the control of Adaptive RED is quite effective for improving steady state performance compared with the original RED, but has little effect on improving the transient performance.

The rest of this thesis is organized as follows. In Section 2, we briefly explain the operation algorithm of RED. In Section 3, we analyze the transient behavior of the RED gateway for various types of TCP connections variations. Several numerical examples are presented to clearly show how control parameters of the RED gateway or system parameters affect the transient performance. In Section 4, we investigate what type of function is appropriate as a packet loss probability function of RED using analytic results in [16, 8]. By presenting numerical examples, we show that, when the packet loss probability function is concave, the transient behavior and the robustness of RED are improved compared with the case of a linear function. Section 5 presents concluding remarks and also discusses open issues.

2 RED (Random Early Detection)

RED has four control parameters called min_{th}, max_{th}, max_p, and q_w [2]. min_{th} and max_{th} are the minimum and the maximum thresholds, respectively. These thresholds are used to determine a packet marking probability, according to which RED randomly discards an arriving packet. Moreover, RED estimates its average queue length, which is calculated from the current queue length and its history using a low-pass filter. Let q and $\overline{q}$ be the current and the average queue length, respectively. When a new packet arrives at the gateway, RED updates its average queue length $\overline{q}$ using the following equation.

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

where q_w is a control parameter which determines the weight of the current queue length on the previously estimated average queue length.

RED then calculates the packet marking probability p_b using the estimated average queue length. Specifically, RED determines the packet marking probability p_b with

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

where max_p is a control parameter which determines the upper bound of the packet marking probability (see Fig. 1). RED does not distinguish each TCP connection; i.e., RED applies the same packet marking probability p_b to all arriving packets. In what follows, Eq. (16) is referred to as the *packet marking function*.

Finally, RED discards the arriving packet with the probability p_a defined by

$$p_a \quad = \quad \frac{p_b}{1 - count \times p_b}, \tag{2}$$

where $count$ is the number of packets that have arrived at the gateway since the last packet discard.

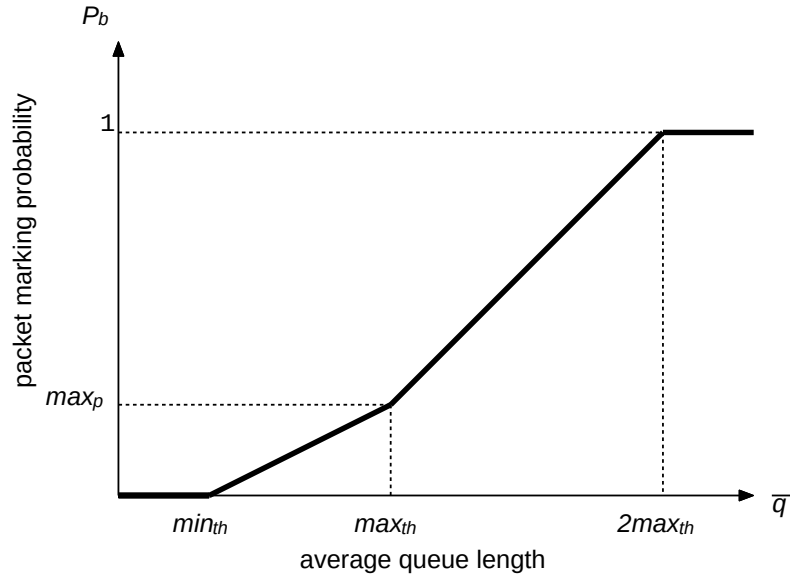

Figure 1: Packet marking function of RED — calculation of packet marking probability p_b from average queue length $\overline{q}$.

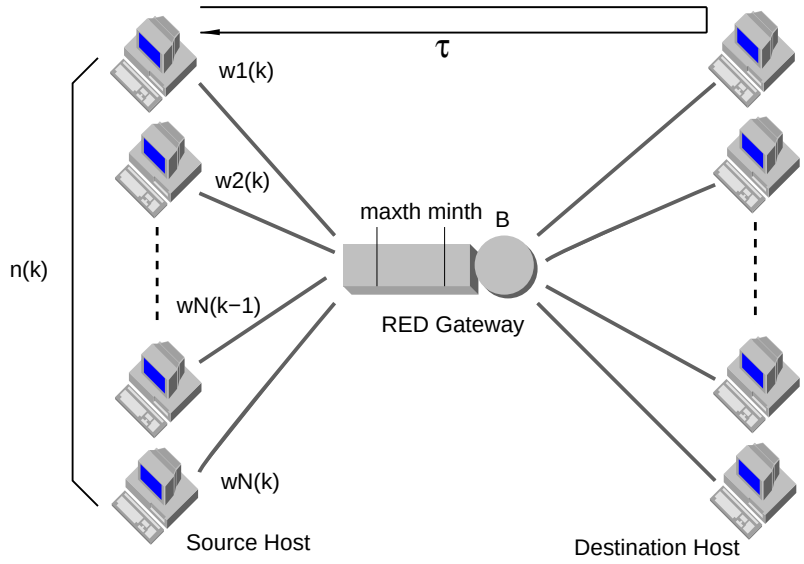

Figure 2: Analytic model.

3 Transient Behavior Analysis of RED using a Control Theoretic Approach

3.1 Analytic Model

In this thesis, we analyze the transient behavior of the RED gateway using the analytic results obtained in [8]. 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 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 [20] 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

Table 1: Definitions of symbols.

min_{th}	minimum threshold value
max_{th}	maximum threshold value
max_p	maximum packet marking probability
q_w	weight factor for averaging
τ	propagation delay of TCP connections
B	processing speed of the RED gateway
$w(k)$	window size at slot k
$q(k)$	current queue length at slot k
$\overline{q}(k)$	average queue length at slot k
$n(k)$	the number of TCP connections at slot k

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 [8]. For taking account of a TCP connections variations, the number of TCP connections at slot k is denoted by $n(k)$. The definitions of symbols are summarized in Tab. 1.

3.2 Average State Transition Equations

In this subsection, we present the derivation of average state transition equations, which describe the dynamics of the RED gateway [8]. Refer to [8] for the detail of the analysis.

3.2.1 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,

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

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

$$w(k+1) = \begin{cases} \frac{w(k)}{2} & \text{with probability } \overline{p} \\ w(k) + 1 & \text{otherwise} \end{cases} \quad (3)$$

Note that in the above equation, it is assumed that all packet losses can be detected by duplicate ACKs [8]. The current queue length at slot $k+1$ is given by

$$\begin{aligned} q(k+1) &= q(k) + n(k+1)\, w(k+1) - B\left(\tau + \frac{q(k)}{B}\right) \\ &= n(k+1)\, w(k+1) - B\,\tau \end{aligned} \tag{4}$$

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

$$\begin{aligned} \overline{q}(k+1) &= (1-q_w)^{n(k)\,w(k)}\overline{q}(k) \\ &\quad + \frac{q_w\{1-(1-q_w)^{n(k)\,w(k)}\}}{1-(1-q_w)}\, q(k) \end{aligned}$$

3.2.2 Derivation of Average State Transition Equations

We derive average state transition equations that represent a typical behavior of TCP connections and the RED gateway [8]. We 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 (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. (3) as

$$w(k+\overline{s}(k)) = \frac{w(k)+\overline{s}(k)-1}{2} \tag{5}$$

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. (4) as

$$q(k+\overline{s}(k)) = n(k+\overline{s}(k))\, w(k+\overline{s}(k)) - B\,\tau \tag{6}$$

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

$$\begin{aligned} \overline{q}(k+\overline{s}(k)) &\simeq (1-q_w)^{\overline{X}(k)}\overline{q}(k) \\ &\quad + \{1-(1-q_w)^{\overline{X}(k)}\} q(k) \end{aligned} \tag{7}$$

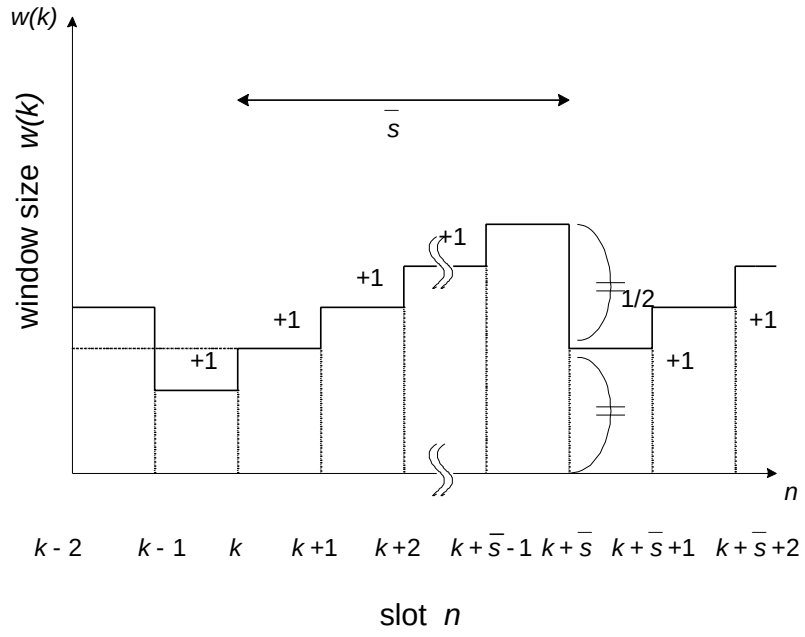

Figure 3: Relationship between slot and sequence.

Average state transition equations given by Eqs. (5), (6), and (7) 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 the 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 $\overline{\mathbf{x}}(k)$ as the difference between the state vector $\mathbf{x}(k)$ and the average equilibrium point.

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

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

$$\overline{\mathbf{x}}(k + \overline{s}(k)) \simeq \mathbf{A}\overline{\mathbf{x}}(k) \tag{8}$$

where **A** is a state transition matrix.

3.3 Transient Behavior Analysis

We assume that N TCP connections exist in the 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 (denoted by C1) is that ΔN ($\Delta N < N$) TCP connections of N TCP connections end their data transmissions. 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 thesis, 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).

1. Cases C1 and C2: Congestion Avoidance Phase Only

   First, the cases C1 and C2, where all TCP connections will operate in the congestion avoidance phase after the TCP connections variation, are considered. Let $u(k)$ be the difference

in the numbers of TCP connections at slot $k-1$ and slot k. Also let $y(k)$ be the queue length of the RED gateway at slot k (Fig. 4). The linearized state transition equations given by Eq. (8) can be extended to include $u(k)$ and $y(k)$ as the input and the output, respectively [21]. Namely,

$$\begin{aligned} \overline{\mathbf{x}}(k+\overline{s}(k)) &= \mathbf{A}\overline{\mathbf{x}}(k) + \mathbf{B}u(k) \\ y(k) &= \mathbf{C}\overline{\mathbf{x}}(k) \\ \mathbf{B} &= [\,0\,0\,0\,1\,]^T \\ \mathbf{C} &= [\,0\,1\,0\,0\,] \end{aligned} \tag{9}$$

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. (10), 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

$$q(k) = \sum_{i=0}^{k} u(i)\,\overline{\mathbf{x}}(k-i) \tag{10}$$

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 [22]. 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.

2. Cases C3 and C4: Congestion Avoidance Phase and Slow Start Phase

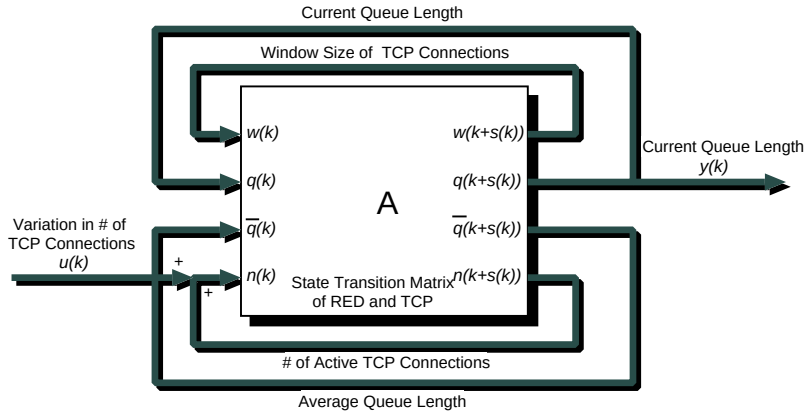

Figure 4: Cases C1 and C2 where all TCP connections in congestion avoidance phase.

Second, the other two cases (C3 and C4), where a part of TCP connections will operate in the slow start phase after the TCP connection variation, are considered. Let $u(k)$ be the difference in the total numbers of packets sent from source hosts being in the slow start phase at slot $k-1$ and slot k. Let $y(k)$ be the queue length of the RED gateway at slot k (Fig. 4). Similar to the previous cases, the linearized state transition equations given by Eq. (8) can be extended to include $u(k)$ and $y(k)$ as the input and the output, respectively.

$$\begin{aligned} \overline{\mathbf{x}}(k+\overline{s}(k)) &= \mathbf{A}\overline{\mathbf{x}}(k)+\mathbf{B}u(k) \qquad (11)\\ y(k) &= \mathbf{C}\overline{\mathbf{x}}(k)\\ \mathbf{B} &= [\,1\,0\,0\,0\,]^T\\ \mathbf{C} &= [\,0\,1\,0\,0\,] \end{aligned}$$

In this thesis, we analyze the transient behavior of the RED gateway for various types of TCP connections variations by utilizing the transfer function. The transfer function of a linear system describes the relation between the input and the output in frequency domain [22]. We define z-transforms of the input $u(k)$ and the output $y(k)$ as $U(z)$ and $Y(z)$, respectively. The transfer

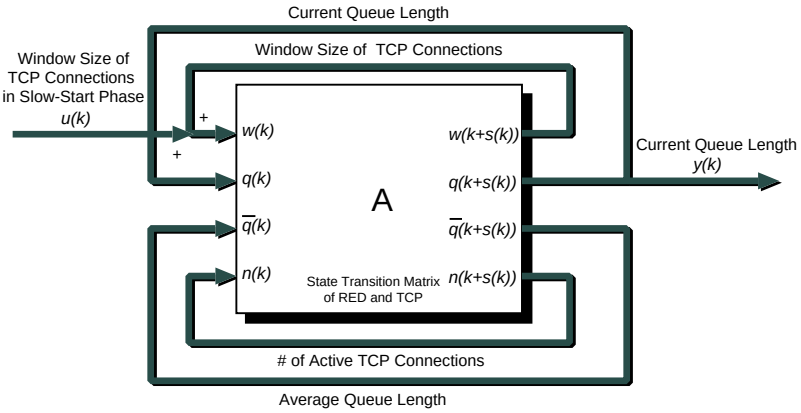

Figure 5: Cases C3 and C4 where a part of TCP connections in slow start phase.

function $G(z)$ of a system satisfies

$$Y(z) \quad = \quad G(z)\,U(z) \tag{12}$$

The transfer function of the system described by Eqs. (9) and (11) is given by the following equation [22].

$$G(z) \quad = \quad \mathbf{C}(z\,\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{B}$$

It is known that the stability and the transient behavior of a closed-loop system are determined by the poles of its transfer function. By investigating the modulus of poles λ_i, the stability and the transient behavior of the system can be easily known. In our transient behavior analysis, the input is the z-transform of either the difference in the numbers of TCP connections (C1 and C2) or the difference in the sum of window sizes of all source hosts (C3 and C4). The transient performance of the RED gateway is therefore determined by poles of $G(z) \times U(z)$. However, Eq. (12) always has the modulus of 1.0 since the number of TCP connections $n(k)$ is changed only by the input $U(z)$. It is therefore necessary to exclude the modulus of 1.0 for analyzing the transient behavior since the pole corresponding to the modulus of 1.0 has no impact on the transient behavior of the queue length of the RED gateway. Note that we have lineally approximated the system around its

equilibrium values in deriving Eq. (8). It is therefore expected that the approximation error becomes large when the value of the window size or the queue length deviates from their equilibrium values. In Subsection 3.4, we therefore validate our approximate analysis by comparing analytic results with simulation ones.

First, we explain examples of the input $U(z)$ in cases C1 and C2, where all TCP connections will operate in the congestion avoidance phase.

1. Case of temporary change in the number of TCP connections

   A temporary change in the number of TCP connections can be formulated by an impulse. For example, when the number of TCP connections is increased by ΔN (or decreased if ΔN is negative) at sequence i, the input $u(k)$ and its z-transform $U(z)$ are given by

$$\begin{aligned} u(k) &= \begin{cases} \Delta N & \text{if } k = S_i \\ 0 & \text{otherwise} \end{cases} \\ U(z) &= \Delta N \, z^{-i} \end{aligned} \tag{13}$$

2. Case of generic change in the number of TCP connections

   A generic change in the number of TCP connections can be given by the convolution of multiple impulses. For instance, when the number of TCP connections is increased (or decreased) by ΔN_i at sequence t_i, the z-transform of the input, $U(z)$, is given by

$$U(z) = \sum_i \Delta N_i z^{-t_i} \tag{14}$$

3. Case of continuous change in the number of TCP connections

   A continuous change in the number of TCP connections can be formulated by the step function. For example, when the number of TCP connections is increased (or decreased) by ΔN at each sequence, the z-transform of the input, $U(z)$, is given by

$$U(z) = \frac{\Delta N \, z}{z - 1}$$

Second, we explain examples of the input $U(z)$ in cases C3 and C4, where a part of TCP connections will operate in the slow start phase.

1. Case of temporary increase in the number of TCP connections

   When a part of TCP connections will operate in the slow start phase, window sizes of these connections are doubled every round-trip time. Hence, when the number of TCP connections in the slow start phase is increased by ΔN at sequence i, $u(k)$ and $U(z)$ are approximately given by

$$\begin{aligned} u(k) &\simeq \begin{cases} \frac{\Delta N}{n(k)} \times 2^{\overline{s}(k)(k-S_i-1)} & \text{if } k > S_i \\ 0 & \text{otherwise} \end{cases} \\ U(z) &\simeq \frac{2^{\overline{s}^*(-S_i-1)}\,\Delta N\, z}{n^*\,(z-2)} \end{aligned} \tag{15}$$

   In the above equation, we approximate as $n^* \equiv n(k)$ since the number $n(k)$ of TCP connections in the congestion avoidance phase does not change. Similarly, we approximate as $\overline{s}^* \equiv \overline{s}(k)$.

2. Case of generic increase in the number of TCP connections

   A generic increase in the number of TCP connections can be formulated by the convolution of Eq. (15). For instance, when the number of TCP connections operating in the slow start phase is increased by ΔN at sequence t_i, z-transform $U(z)$ is given by

$$U(z) = \sum_i \frac{\Delta N_i\, 2^{\overline{s}^*(-i-1)}\, z^{-t_i+1}}{n^*\,(z-2)}$$

3.4 Numerical Examples and Discussions

In this section, presenting several numerical examples, we discuss the relation between a choice of control parameters of the RED gateway and its transient behavior. We also validate our approximate analysis presented in subsection 3.1 by comparing analytic results with simulation ones. In all analytic and simulation results, without explicitly stated, the following parameters are used: the

number of TCP connections in the steady state $n^* = 5$, the processing speed of the RED gateway $B = 0.2$ [packet/ms], the propagation delay $\tau = 1$ [ms]. Four control parameters of the RED gateway are configured according to the recommendation in [2], i.e., $min_{th} = 5$, $max_{th} = 15$, $max_p = 0.1$ and $q_w = 0.002$. We note that with these control parameters RED operates successfully; i.e., the average queue length is stabilized between min_{th} and max_{th} [2].

3.4.1 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. 6) [23]. 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 thesis, 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 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. (4), the current queue length, $q(k)$, directly reflects the window sizes $w(k)$. So it

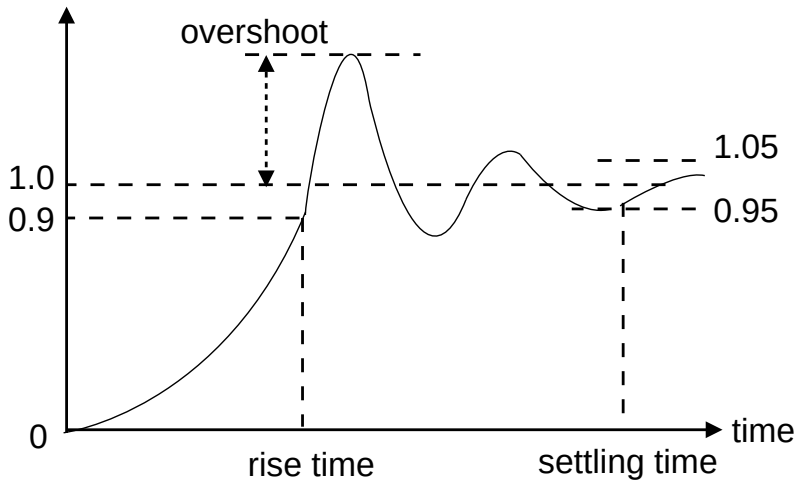

Figure 6: Performance measures for transient behavior (overshoot, rise time, and settling time).

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.

3.4.2 Overshoot, Rise Time, and Settling Time

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. (12) when ΔN TCP connections resume at slot 0; i.e.,

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

Note that the dynamics of the queue length $q(k)$ of the RED gateway can be obtained from direct calculation using Eq. (12).

Figure 7 shows performance measures for the transient behavior (the overshoot, the rise time,

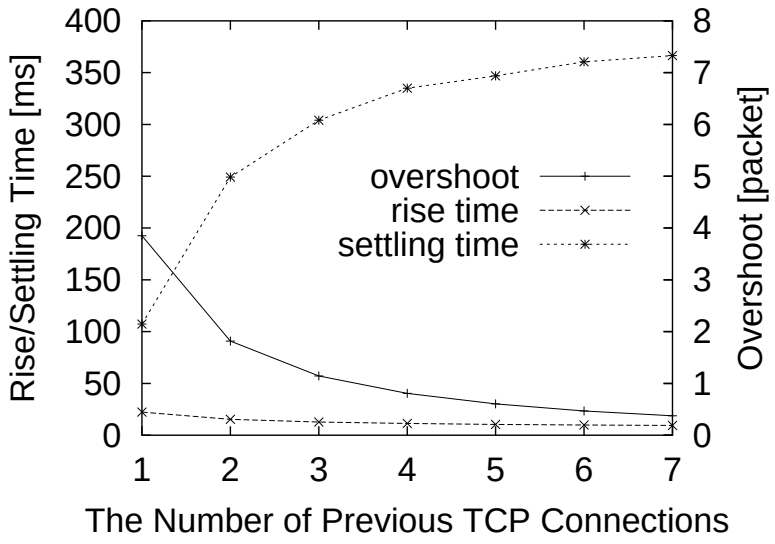

Figure 7: Performance measures for transient behavior (the number of previous TCP connections N = 1–7).

and the settling time) for different number N of TCP connections in the 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 [2]. We also use the following system parameters: the processing speed of the RED gateway $B = 0.2$ [packet/ms], the propagation delay $\tau = 1$ [ms], and the number of resumed TCP connections $\Delta N = 1$. Figure 7 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 the 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 overflow at the RED gateway is not likely to happen when the number of TCP connections is sufficiently large.

Shown in Fig. 8 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

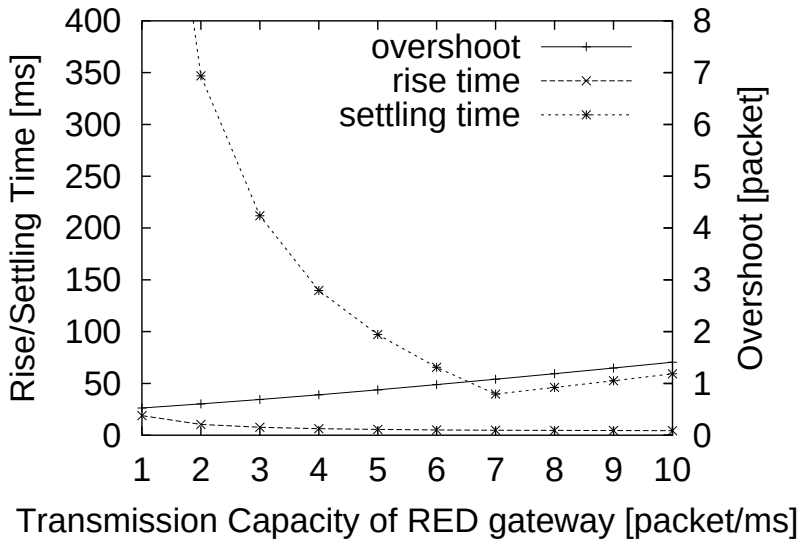

Figure 8: Performance measures for transient behavior (the processing speed of the RED gateway B = 1–10 [packet/ms])

decreases, the overshoot and the settling time become 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 9 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 9 also shows that the settling time is minimized when the propagation delay of the TCP

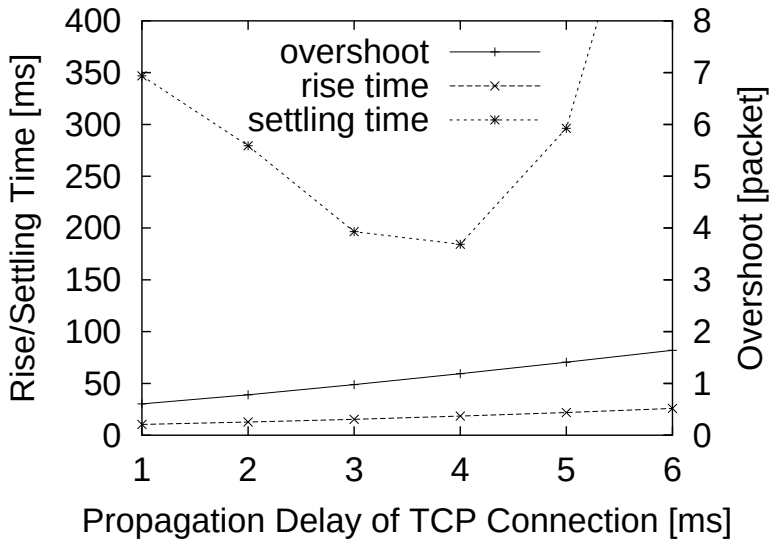

Figure 9: Performance measures for transient behavior (the propagation delay of the TCP connection τ = 1–5 [ms])

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 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 10 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

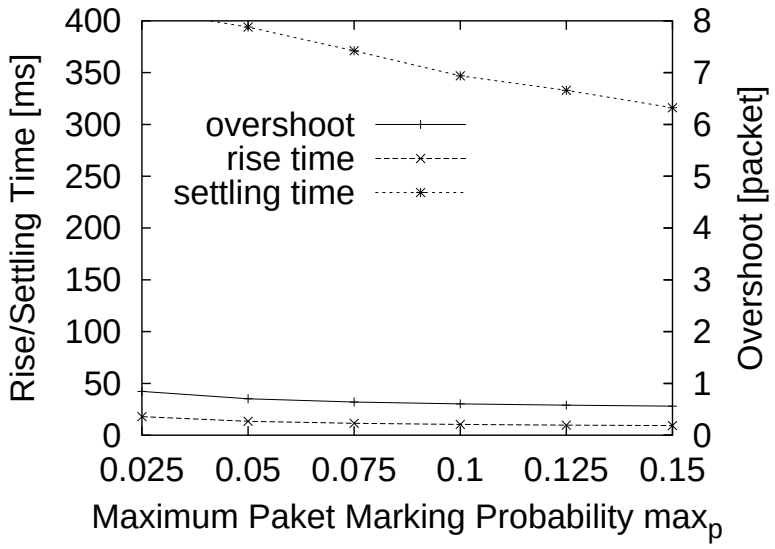

Figure 10: Performance measures for transient behavior (the maximum packet marking probability max_p =0.025–0.15)

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.

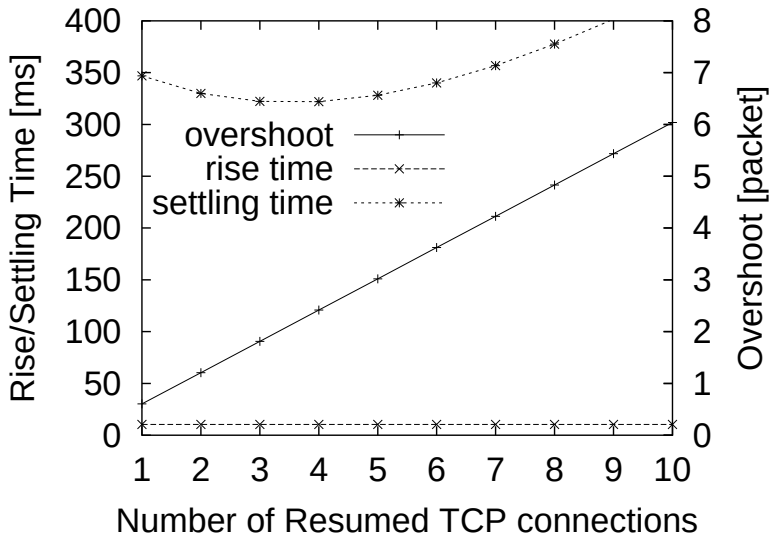

Figure 11: Performance measures for transient behavior (the number of resumed TCP connections ΔN = 1–10)

3.4.3 Stability Analysis

Figure 12 shows the stability region when the number of TCP connections, operating in the congestion avoidance phase, is incremented by one. This figure shows the maximum modulus of poles of $U(z) \times G(z)$ (Eq. (12)) in the min_{th}–max_{th} plane when the input $U(z)$ is given by Eq. (13) with $i = 0$. This figure means that the operation of the RED gateway becomes unstable (i.e., the queue length of the RED gateway never converges) when the maximum modulus of pole is larger than 1.0. This figure also means that the smaller the maximum modulus of pole is, the better the transient behavior of the RED gateway becomes. One can find from this figure that the control parameter min_{th} has a large impact on the transient behavior of the RED gateway. For instance, in this case, the transient behavior of the RED gateway is optimal if min_{th} is chosen around 8.0. On the contrary, one can also find that the control parameter max_{th} has little impact on the stability and the transient behavior. Note that similar tendency is discovered in [8], where the steady state analysis of the RED gateway is performed. From these observations, we can conjecture that if con-

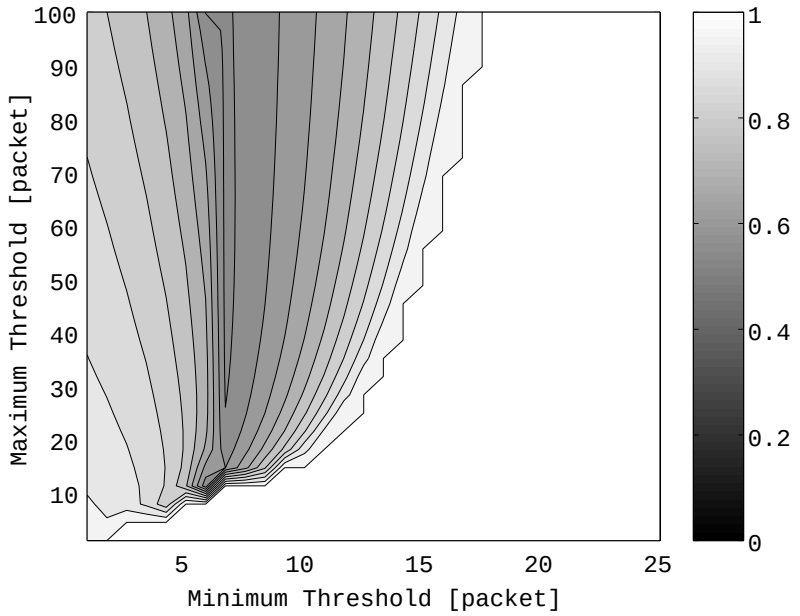

Figure 12: The maximum modulus of poles in the min_{th}–max_{th} plane.

trol parameters of the RED gateway are configured for optimizing its steady state performance, the transient behavior is also optimized.

In Fig. 13, we next show the stability region for $min_{th} = 5$ and $max_{th} = 15$. In this figure, the maximum modulus of poles of $U(z) \times G(z)$ is plotted in the B–τ plane. This figure indicates that both the processing speed of the RED gateway B and the propagation delay τ affect the transient behavior of the RED gateway. This figure also indicates that the maximum modulus of poles are mostly determined by the bandwidth–delay product ($B \times \tau$).

3.4.4 Transient Behavior Analysis using Transfer Function

Figures 14 through Fig. 16 present numerical results from our transient behavior analysis based on the transfer function. These figures are for the cases that the number of TCP connections, operating in the congestion avoidance phase, is increased. In all these figures, Eq. (14) is used as the input $U(z)$, but t_i's and ΔN_i's are set to different values. In Fig. 14, $t_1 = 5$ and $\Delta N_1 = 1$

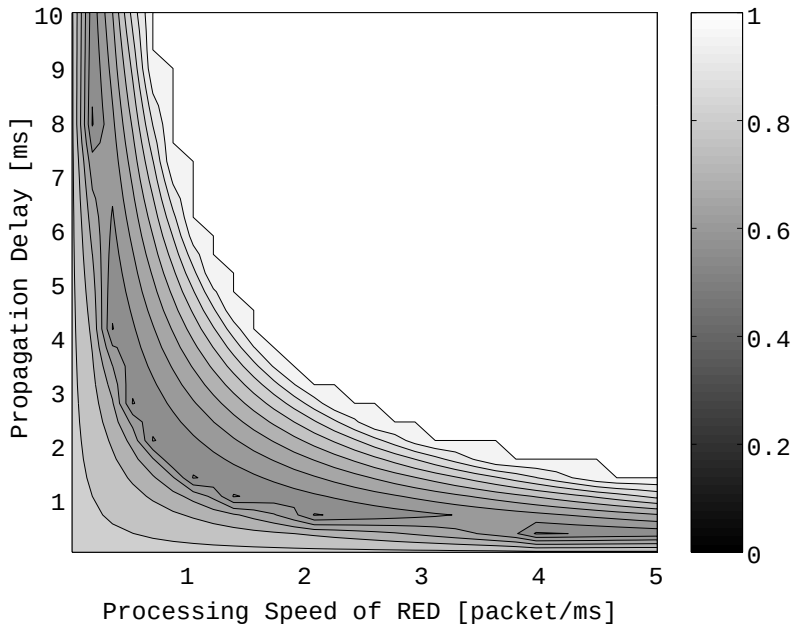

Figure 13: The maximum modulus of poles in the B–τ plane.

are used for analyzing the case that the number of TCP connections is incremented by one. In Fig. 15, $(t_1, t_2, t_3) = (5, 10, 15)$ and $(\Delta N_1, \Delta N_2, \Delta N_3) = (1, 1, 1)$ are used for analyzing the case that the number of TCP connections is incremented by three. Finally, in Fig. 16, $(t_1, t_2, t_3) = (5, 10, 15)$ and $(\Delta N_1, \Delta N_2, \Delta N_3) = (1, 2, 1)$ for analyzing the case that the number of TCP connections is incremented by four.

In each figure, the impulse response of the transfer function $U(z) \times G(z)$ (upper) and the gain characteristic of $U(z) \times G(z)$ (lower). The impulse response illustrates the evolution of the queue length of the RED gateway. The gain characteristic illustrates the amplitude of the transfer function $U(z) \times G(z)$ at different frequencies. Note that these impulse responses are obtained not from iterative computation using Eq. (8), but from direct calculation using the transfer function and the MATLAB language. By analyzing these impulse responses, we can investigate the transient behavior of the RED gateway for various types of TCP connections variations. For example, comparing gain characteristics of Figs. 15 and 16 tells us that the amplitude of the queue length

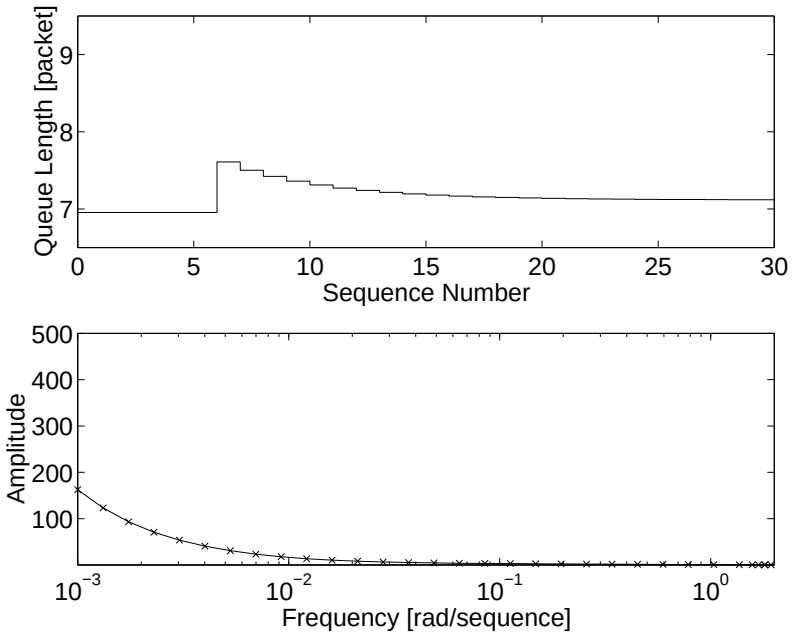

Figure 14: Impulse response and gain characteristic of $U(z) \times G(z)$ for $\Delta N_1 = 1$.

of the RED gateway in Fig. 16 is larger than that in Fig. 15. In other words, using the transfer function enables us to analyze the transient behavior of the RED gateway not in time domain but in frequency domain.

Finally, in Fig. 17, we show a simulation result when all TCP connections operate in the congestion avoidance phase (i.e., case C2). This figure shows the evolution of the queue length when the number of TCP connections operating in the congestion avoidance phase is incremented by one at $t = 10$ [s]. In this simulation, we use the ns-2 simulator for the same network model with Fig. 2. The size of all TCP packets is set to 1,000 [bytes]. In this figure, the analytic result is also plotted for comparison purposes. This figure shows that the analytic result roughly coincides to the simulation result. In particular, after the change in the number of TCP connections, the settling time of the queue length in simulation is almost identical to the analytic result. However, one can find that the amplitude of the queue length in our analysis is smaller than that of simulation. Such difference might be caused by approximation errors in linearization or conservative estimation of

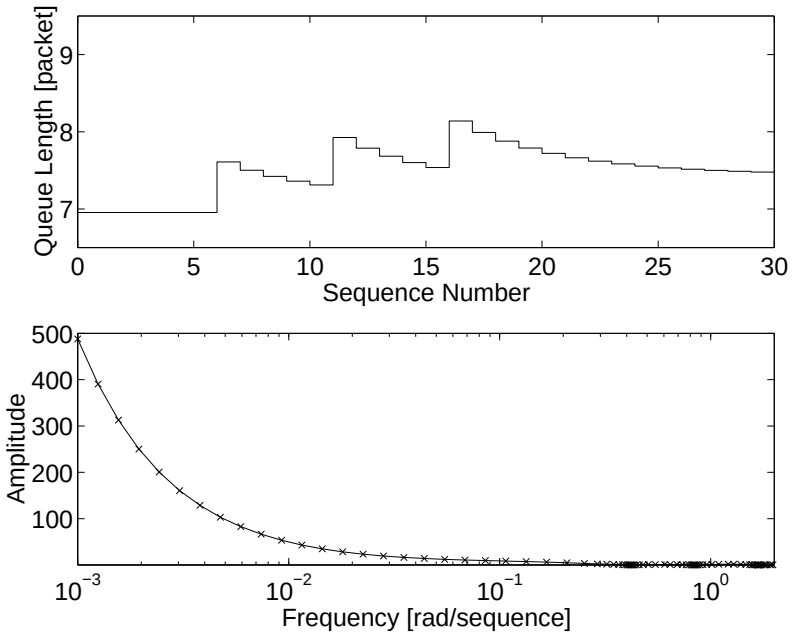

Figure 15: Impulse response and gain characteristic of $U(z) \times G(z)$ for $(\Delta N_1, \Delta N_2, \Delta N_3) = (1, 1, 1)$.

$w(k)$ or $q(k)$. As a future work, more investigation for minimizing the difference between analytic and simulation results would be appropriate.

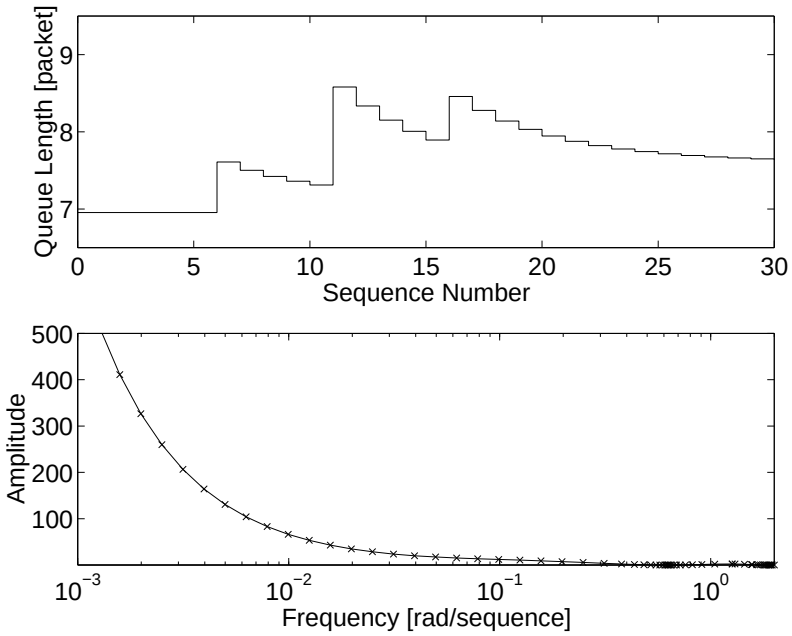

Figure 16: Impulse response and gain characteristic of $U(z) \times G(z)$ for $(\Delta N_1, \Delta N_2, \Delta N_3) = (1, 2, 1)$.

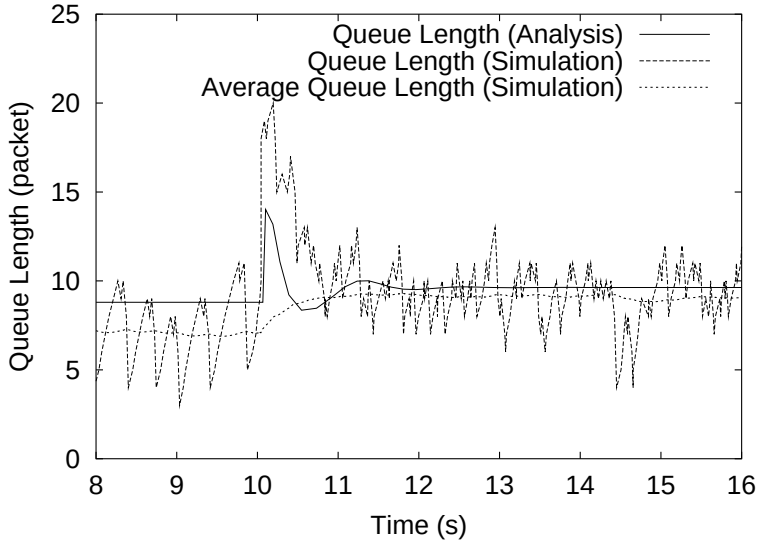

Figure 17: Comparison between analytic and simulation results.

4 Packet Marking Function of Active Queue Management Mechanism: Should It Be Linear, Concave, or Convex?

4.1 Analysis

In what follows, we consider the case when the AQM mechanism of RED is operating as expected, i.e., $min_{th} \leq \overline{q} < max_{th}$ because control parameters of the RED gateway are configured appropriately. In this case, the packet marking probability p_b is determined based on the average queue length $\overline{q}$ as

$$p_b \quad = \quad max_p \left(\frac{\overline{q} - min_{th}}{max_{th} - min_{th}} \right). \tag{16}$$

When the average queue length $\overline{q}$ is large, p_b takes a value close to max_p. On the contrary, when the average queue length $\overline{q}$ is small, p_b takes a value close to zero. As can be seen from Eq. (16), the packet marking function of RED is a linear function in relation to $(\overline{q} - min_{th})$. However, use of a linear function has not been fully validated by taking account of both steady-state and transient-state performances of RED. For example, the window-based flow control of TCP changes its window size non-linearly according to the packet loss probability in the network. For this reason, if the packet marking probability p_b is determined by taking account of the characteristic of the TCP window-based flow control, the steady state performances and/or transient state performances of RED can be improved. In what follows, we therefore discuss how the packet marking probability p_b should be determined by utilizing the analytic results of TCP [16] and RED [8].

In our analysis, by combining the stochastic model of the TCP window size and the deterministic model of the RED queue length, we analyze how the average queue length is affected by the RED's packet marking mechanism. Specifically, we analyze toward what value the average queue length converges with the packet marking mechanism of RED when the average queue length is given at some time. Consequently, we clarify the effect of the packet marking function on the average queue length of RED in steady state (i.e., steady state performance) the average queue length

of RED in transient state (i.e., transient state performance), and robustness against variation in the number of active TCP connections.

First, the packet marking function of RED as defined by Eq. (16) is replaced by

$$p_b \quad = \quad max_p \, f\left(\frac{\overline{q} - min_{th}}{max_{th} - min_{th}}\right), \tag{17}$$

where f is a monotonically increasing function satisfying $f(0) = 0$ and $f(1) = 1$. We then introduce the notion of *queue occupancy* to indicate how many packets exist in the gateway's buffer. The queue occupancy is defined by

$$x \equiv \frac{\overline{q} - min_{th}}{max_{th} - min_{th}}$$

for $min_{th} \leq \overline{q} < max_{th}$. The purpose of the remainder of this subsection is to investigate how the packet marking function f affects the steady state and transient state performances of RED.

We consider the expected value of the TCP window size when the packet loss probability in the network is given. In [16], the TCP throughput in steady state is derived. In the derivation process, $W(p)$, which is the expected value of TCP window size just before TCP detects a packet loss, is also derived (see Eq. (18) in [16]).

$$W(p) \quad = \quad \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \tag{18}$$

In the above equation, b is the number of packets required for the destination host to return an ACK packet (usually, $b = 1$ or $b = 2$), and p is the packet loss probability in the network. In the analysis, the authors assume (1) TCP is operating in the congestion evasion phase, (2) all packet losses are detectable by duplicate ACKs (i.e., no timeout is triggered), (3) the packet loss probability in the networks is constant, and (4) the maximum window size is sufficiently large. Note that in [16], the second assumption (i.e., no timeouts) is relaxed and the more detailed result of the TCP window size is derived. However, we use a simple result given by Eq. (18) since it is more tractable than the detailed result, allowing us to know more insight on the packet marking function of RED. Also, note that Eq. (18) gives the expected value of TCP window size just before a packet loss is

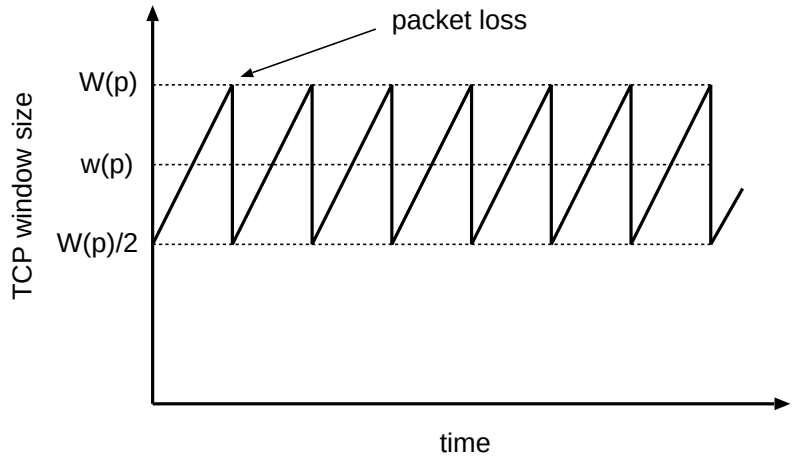

Figure 18: Evolution of TCP window size in congestion avoidance phase.

detected. Immediately after detecting the packet loss, the TCP window size is decreased to one half. Then, the TCP window size is slowly increased until the next packet loss is detected. In this thesis, we therefore use $w(p)$ as the expected value of TCP window size (Fig. 18):

$$w(p) \quad = \quad \frac{1}{2}\left(\frac{W(p)}{2} + W(p)\right) = \frac{3\,W(p)}{4}. \tag{19}$$

Furthermore, we consider the queue length of RED in steady state when the TCP window size w is given. In [8], the queue length $\overline{q}$ of RED is derived when the number N of TCP connections have the identical window size w.

$$\overline{q} \quad = \quad N\,w - B\,\tau, \tag{20}$$

where B is the maximum transmission capacity of the RED gateway (i.e., the smaller value between the processing speed of the RED gateway and the bandwidth of the outgoing link). τ is the two-way propagation delay of the TCP connection excluding the queueing delay at the buffer. In the analysis, almost the same assumptions as those of [16] are made.

As we have explained in Section 2, RED randomly discards an arriving packet with probability p_b. Hence, for a given packet marking probability p_b, the packet loss probability to TCP

connections is given by [2]

$$p = \left(\frac{1}{2p_b} + \frac{1}{2}\right)^{-1}. \quad (21)$$

From Eqs. (17), (20), and (21), the average queue length $\overline{q}$ can therefore be given by

$$\begin{aligned} \overline{q} &= N\,w(p) - B\,\tau \\ &= \frac{N}{4\,b}\left\{2 + b + b\sqrt{\frac{4 - 8\,b + b^2 + \frac{12\,b}{max_p\,f(x)}}{b^2}}\right\} - B\,\tau. \end{aligned} \quad (22)$$

This equation indicates that the queue length of RED converges to $\overline{q}$, when the packet marking probability p_b is determined according to the function f. Note that $\overline{q}$ is the expected value of the queue length since Eq. (19) is also the expected value. Let x^* be the queue occupancy converged by the packet dropping mechanism of RED (i.e., the average queue length in steady state if p_b is fixed). x^* is given by

$$x^* = \frac{\overline{q}(x) - min_{th}}{max_{th} - min_{th}}. \quad (23)$$

Namely, this means that the probabilistic packet marking mechanism of RED with the packet marking function f and the queue occupancy x governs the queue occupancy toward x^*. For example, by plotting Eq. (23) on the x–x^* plane, the effect of the packet marking function f on the steady state and transient state performances of RED can be rigorously analyzed.

To analyze the relation between Eq. (23) and RED performance, we graphically plot the relation between queue occupancies x and x^*. An example of Eq. (23) is shown in Fig. 19. The straight line $x^* = x$ is also plotted in the figure. The following points can be inferred from this figure.

1. The queue occupancy in steady state ($t \to \infty$) is given by the intersection of the curve of Eq. (23) and the straight line $x^* = x$.

2. The steeper the gradient (dx^*/dx) of Eq. (23), the larger the impact of variation in the queue occupancy x on the average queue length of RED.

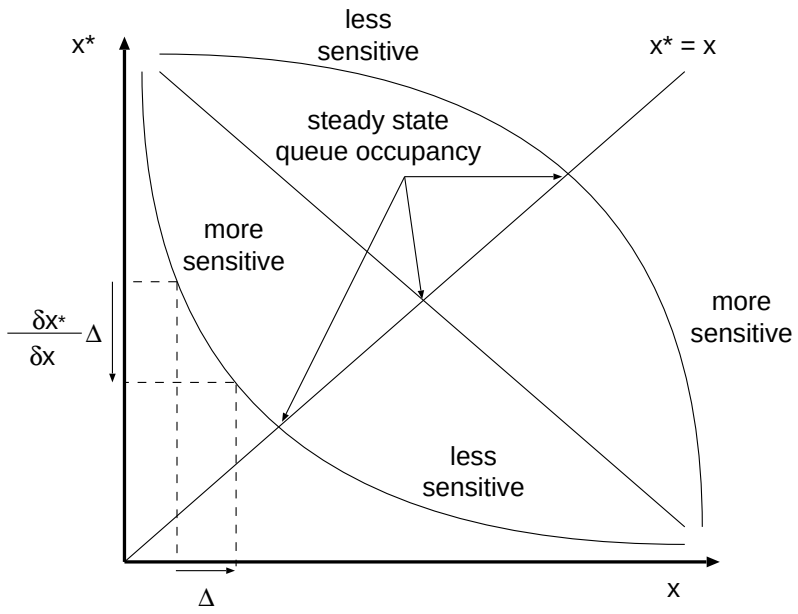

Figure 19: Queue occupancy in the x-x^* plane.

3. Conversely, when the gradient (dx^*/dx) of Eq. (23) is gentle, variation in the queue occupancy x has a negligible effect on the average queue length of RED.

4. In order for the average queue length of RED to be stable when $min_{th} < \overline{q} < max_{th}$, the gradient ($dx^*/dx$) of Eq. (23) must be negative.

On the basis of these observations, we find the following points with regard to the steady state and transient state performances of RED.

1. When Eq. (23) is convex (i.e., $d^2x^*/dx^2 < 0$), the average queue length of RED in steady state becomes small. Moreover, when the queue occupancy is small, transient state performance is good. Conversely, when the queue occupancy is large, transient state performance becomes degraded.

2. When Eq. (23) is linear (i.e., $d^2x^*/dx^2 = 0$), the average queue length of RED in steady state is larger than that with a convex function. Also, transient state behavior is not affected

by the queue occupancy.

3. When Eq. (23) is concave (i.e., $d^2x^*/dx^2 > 0$), the average queue length of RED in steady state is large, and the transient state behavior is not good when the queue occupancy is small. On the contrary, transient state performance is good when the queue occupancy is large.

On the basis of the above observations, it is considered that the packet marking function f should be chosen such that Eq. (23) becomes a linear function in relation to x. Namely, it is desirable for the gradient $d\overline{q}/dx$ of Eq. (22) given by the following equation to be constant.

$$\frac{d\overline{q}}{dx} = \frac{-3\,N\,f'(x)}{2\,max_p\sqrt{4-8\,b+b^2+\frac{12\,b}{max_p\,f(x)}}\,f(x)^2} \tag{24}$$

This equation suggests that the gradient of Eq. (22) is determined by the parameter at the TCP destination host b, the number of active TCP connections N, and the maximum packet marking probability max_p. Conversely, it suggests that the gradient of Eq. (22) is almost independent of the propagation delay of TCP connections τ and the capacity of the RED gateway B. This result agrees with conventional research results that the average queue length $\overline{q}$ of RED is dependent on the number of TCP connections [8, 14, 6]. Therefore, when considering how the packet marking function f should be determined, the number of active TCP connections N should be primarily taken account of.

Moreover, the packet marking function f that makes Eq. (23) a linear function in relation to x can be derived by equating $d\overline{q}/dx$ from Eq. (24) with a constant value α and by solving this ordinary differential equation. The solution is given by

$$f(x) = 12\left\{max_p\left(8-\frac{4}{b}+b\left(\frac{16\,\alpha^2\,x^2}{N^2}-1\right)\right)-\frac{48\,\alpha\,b\sqrt{max_p}\,x\,C(1)}{N}+36\,b\,C(1)^2\right\}^{-1},$$

where $C(1)$ is a constant. As can be seen from the above equation, for determining f so that Eq. (22) becomes linear, f must be changed according to the number of active TCP connections N. Our analytic results clearly indicates that information on the number of active TCP connections is necessary for determining the packet marking probability p_b to optimize the steady state and

transient state performances of RED. However, the RED gateway has no capability of knowing the number of active TCP connections since it does not distinguish each TCP connection. Of course, it is not impossible to estimate the number of active TCP connections, for example, by distinguishing each TCP connection as is done in SRED. However, adding such processing to the AQM gateway makes its implementation very complex. Since implementation simplicity is one of important features of RED, in reality, it is desirable to choose a packet marking function f such that f is as independent of the number of TCP connections as possible, and that Eq. (23) has as much linearity in relation to x as possible.

Hence, in the next subsection, we consider three types of packet marking functions — linear, concave, and convex —and examine which packet marking function is most suitable for optimizing RED performance. Note that the actual steady state and transient state performances of RED such as throughput is determined by not only the type of packet marking function but also the setting of its control parameters such as min_{th}, max_{th}, max_p, and q_w. However, in what follows, we limit our attention to the packet marking function, and carefully investigate how the packet marking function affects the steady state and transient state performances of RED.

4.2 Numerical Examples and Discussions

4.2.1 Case of RED (Random Early Detection)

In the following numerical examples, three types of function classes, $\mathcal{F}_\phi$, $\mathcal{G}_\phi$, and $\mathcal{H}_\phi$, are considered as the packet marking function f.

- Linear

$$\mathcal{F}_\phi(x) \quad = \quad x^\phi$$

  Note that this function is linear when $\phi = 1$, is concave when $\phi > 1$, and is convex when $\phi < 1$.

- Concave

$$\mathcal{G}_\phi(x) \;=\; \left(1-\sqrt{1-x^2}\right)^\phi$$

$\phi(>0)$ is a parameter determining concavity. Namely, when ϕ is large, $\mathcal{G}_\phi(x)$ takes a small value. In order for $\mathcal{G}_\phi$ to be concave,

$$\frac{d^2\mathcal{G}_\phi(x)}{dx^2} \;=\; \frac{1}{(1-x^2)^{\frac{3}{2}}}\left\{\phi\left(1-\sqrt{1-x^2}\right)^{\phi-2}\times\left(1+\sqrt{1-x^2}\,\left((\phi-1)\,x^2-1\right)\right)\right\}\geq 0$$

must be satisfied. By solving the above inequalities for ϕ, we have

$$\phi \;\geq\; \lim_{x\to 0}\frac{-1+\sqrt{1-x^2}+x^2\sqrt{1-x^2}}{x^2\sqrt{1-x^2}}=\frac{1}{2}.$$

- Convex

$$\mathcal{H}_\phi(x) \;=\; \left(\sqrt{1-(1-x)^2}\right)^\phi$$

$\phi(>0)$ is a parameter determining convexness. Namely, when ϕ is large, $\mathcal{H}_\phi(x)$ takes a small value. In order for $\mathcal{H}_\phi$ to be convex,

$$\frac{d^2\mathcal{H}_\phi(x)}{dx^2} \;=\; \phi\left(-\left((x-2)\,x\right)\right)^{\frac{\phi-4}{2}}\times\left(\phi\,(x-1)^2-(x-2)\,x-2\right)\leq 0$$

must be satisfied. Similar to the previous case, by solving the above inequalities for ϕ, we have

$$\phi \;\leq\; \lim_{x\to 0}\frac{2-2\,x+x^2}{1-2\,x+x^2}=2.$$

Note that $\mathcal{F}_\phi(x)$ with $\phi=1.0$ is identical to Eq. (16) of the original RED.

We investigate which packet marking function is the most desirable among $\mathcal{F}_\phi$ (linear), $\mathcal{G}_\phi$ (concave), and $\mathcal{H}_\phi$ (convex) by showing several numerical examples. First, the queue occupancy in the x–x^* plane is shown when control parameters and system parameters are configured according to Tab. 2. Figures 20, 21, and 22 respectively show results when $\mathcal{F}_\phi$ (linear), $\mathcal{G}_\phi$ (concave), and

Table 2: Parameters used in numerical examples.

min_{th}	RED maximum threshold	50	[packet]
max_{th}	RED minimum threshold	100	[packet]
max_p	RED maximum packet marking probability	0.1	
B	transmission capability of RED gateway	1.25	[packet/ms]
τ	two-way TCP propagation delay	10	[ms]
N	the number of active TCP connections	10	
b	TCP destination host parameter	1	

$\mathcal{H}_\phi$ (convex) are used as the packet marking function. In these figures, ϕ is changed to 0.5, 1.0, and 1.5. The straight line $x = x^*$ is also plotted in all figures.

Figure 20 shows that the average queue length of RED (i.e., the intersection with the straight line $x = x^*$) increases as the value of ϕ becomes large when $\mathcal{F}_\phi$ (linear) is used as the packet marking function f. Moreover, it is also found that when the queue occupancy x is small, the gradient (dx^*/dx) of Eq. (23) is steep. On the other hand, when the queue occupancy is large, the value of Eq. (23) is almost zero. Namely, in the state where the queue occupancy is small, since RED discards arriving packets, the queue length is changed rapidly. In the state where the queue occupancy is large, the queue length rapidly decreases to min_{th} (i.e., the queue occupancy is empty) irrespective of the packet marking probability of RED.

When $\mathcal{G}_\phi$ (concave) is used as the packet marking function, the gradient (dx^*/dx) of Eq. (23) is small (see Fig. 21). This figure indicates that as the value of ϕ becomes larger (i.e., with increasingly stronger concavity), the average queue length of RED becomes larger. Moreover, this figure shows that Eq. (23) is closer to the straight line as compared with Fig. 20. Namely, the effect that packet losses have on the queue length of RED is almost independent of the queue occupancy.

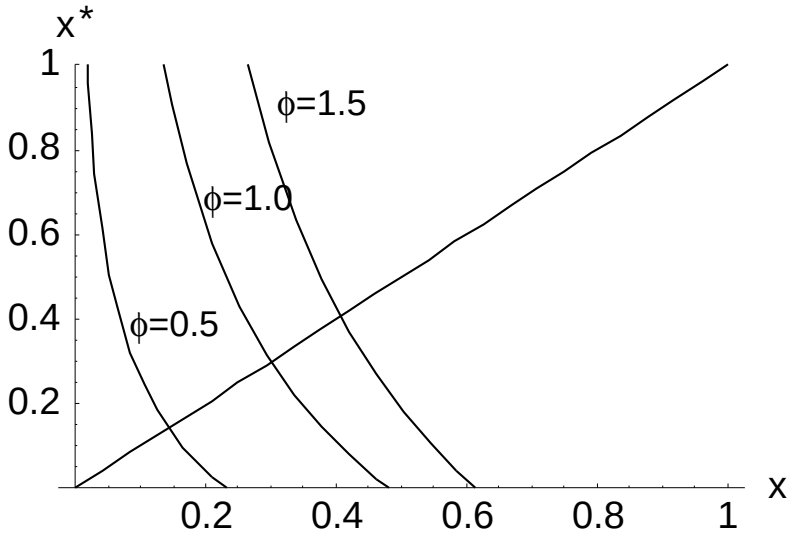

Figure 20: RED queue occupancy in the x–x^* plane with $\mathcal{F}_\phi$ (linear) for ϕ = 0.5, 1.0, and 1.5.

The result when using $\mathcal{H}_\phi$ (convex) as the packet marking function f is shown in Fig. 22. This figure shows the relation of the queue occupancy in the x–x^* plane. It is evident that the gradient (dx^*/dx) of Eq. (23) is quite steep when the queue occupancy is small. Moreover, it is also evident that, as the value of ϕ becomes large (i.e., with increasingly strong convexity), the gradient of Eq. (23) becomes even steeper. Namely, when $\mathcal{H}_\phi$ (convex) is used as the function f, the queue length of RED is changed rapidly when the queue occupancy is small.

Next, when $\mathcal{F}_\phi$ (linear), $\mathcal{G}_\phi$ (concave), and $\mathcal{H}_\phi$ (convex) are used as the packet marking function f, we show the effect of variation in the number of active TCP connections on the steady state performance and transient state performance of RED. The relation between the number of TCP connections N and the queue occupancy when using $\mathcal{F}_\phi$ (linear) is shown in Fig. 23. In this figure, the number of TCP connections N is varied from 1 to 20, and the parameter values shown in Tab. 2 are used. As discussed in subection 4.1, the intersection of the curved surface and the x–x^* plane means the average queue length of RED in steady state. This indicates that the average queue length of RED in steady state becomes larger as the number of TCP connections N

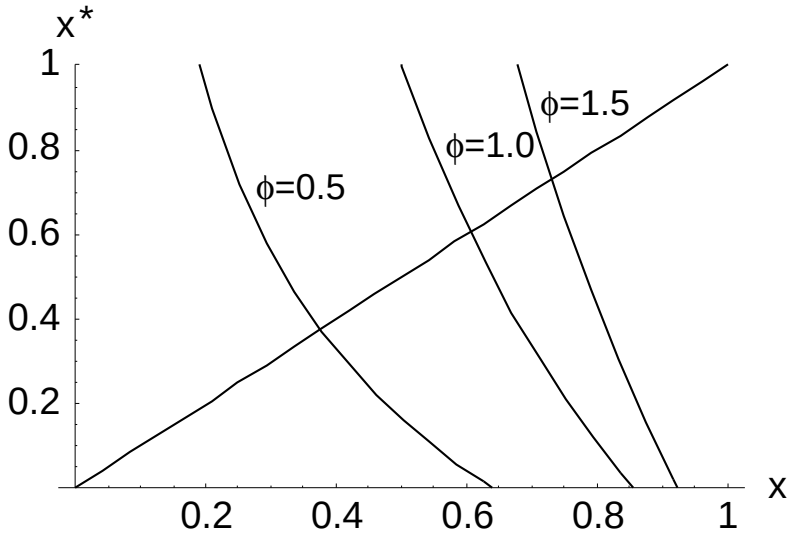

Figure 21: RED queue occupancy in the x–x^* plane with $\mathcal{G}_\phi$ (concave) for ϕ =0.5, 1.0, and 1.5.

becomes large. In particular, it is evident that when the number of TCP connections N is small, the variation in the number of TCP connections significantly affects the average queue length. Furthermore, it is also evident that the gradient (dx^*/dx) of Eq. (23) becomes small as the number of TCP connections N becomes large. This means that the transient state performance of RED is affected by changing the number of TCP connections.

Figure 24 shows the result when using $\mathcal{G}_\phi$ (concave) as the packet marking function f. Except for using $\mathcal{G}_\phi$ (concave) as the packet marking function f, the same parameter values as those for Fig. 23 are used. This figure shows that the gradient (dx^*/dx) of Eq. (23) is negligibly dependent on the number of active TCP connections when $\mathcal{G}_\phi$ (concave) is used. Furthermore, similar to Fig. 23, the average queue length of RED in steady state becomes larger as the number of active TCP connections N becomes large. However, the average queue length of RED is observed to increase almost linearly with the number of TCP connections N. Generally, the number of TCP connections changes according to time. For this reason, Fig. 24 would be more desirable than Fig. 23 in the sense that the average queue length does not change excessively with the number of

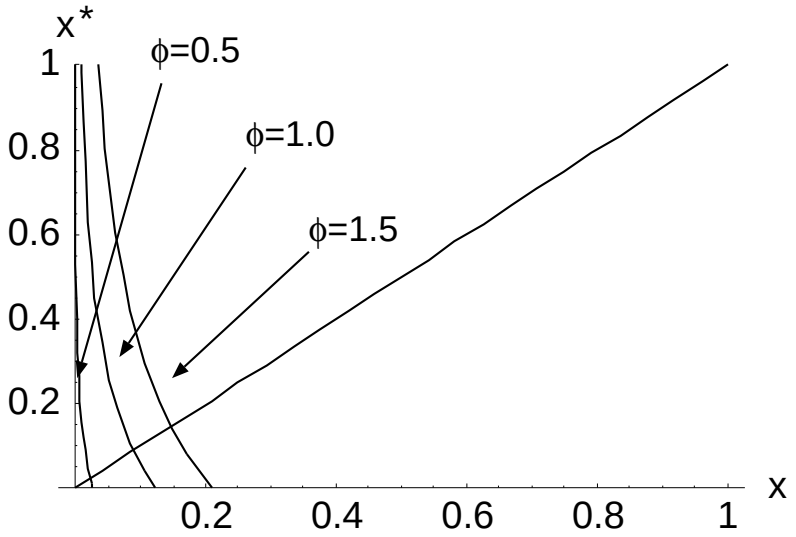

Figure 22: RED queue occupancy in the x–x^* plane with $\mathcal{H}_\phi$ (convex) for ϕ = 0.5, 1, and 1.5.

TCP connections.

Finally, the result when using $\mathcal{H}_\phi$ (convex) as the packet marking function f is shown in Fig. 25. This figure shows that the average queue length in steady state becomes large as the number of TCP connections N becomes large. Moreover, it indicates that the gradient (dx^*/dx) of Eq. (23) is significantly influenced by the number of TCP connections. For this reason, considering the steady state behavior and the transient state behavior of RED, we consider that $\mathcal{H}_\phi$ (convex) is unsuitable for application as the function f.

From the above observations, we conclude that $\mathcal{G}_\phi$ (concave) is the most suitable for application as the packet marking function f. Although the average queue length in steady state can be small when either $\mathcal{F}_\phi$ (linear) or $\mathcal{H}_\phi$ (convex) is used, there is a drawback that the transient state performance is quite sensitive to the queue occupancy. On the other hand, when $\mathcal{G}_\phi$ (concave) is used, variation in the queue occupancy and the number of TCP connections negligibly affects the transient state behavior of RED. In summary, when $\mathcal{G}_\phi$ (concave) is used as the packet marking function, the transient state performance of RED and robustness to variations in the number of

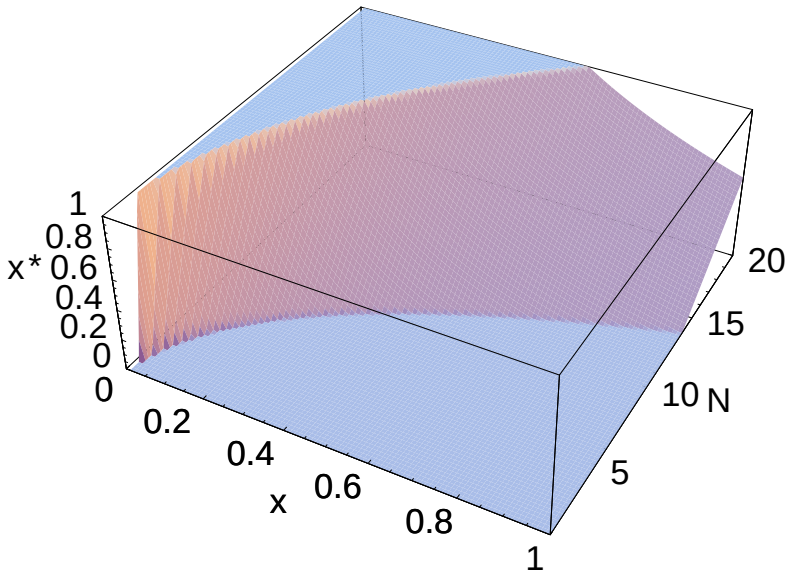

Figure 23: Relation between the number of active TCP connections N and RED queue occupancy with $\mathcal{F}_\phi$ (linear) for ϕ =1.0.

active TCP connections are improved.

4.2.2 Case of Adaptive RED

As we have explained in Section 1, Adaptive RED solves the problem associated with RED that its average queue length is dependent on the number of active TCP connections [19]. Adaptive RED adaptively changes its maximum packet marking probability max_p according to its average queue length. More specifically, when the average queue length is smaller than min_{th}, max_p is decreased by $(1/\alpha - 1)\, max_p$. Conversely, when the average queue length is larger than max_{th}, max_p is increased by $(1-\beta)\, max_p$. The main purpose of the Adaptive RED control is to improve the steady state performance of RED. However, it has not been fully investigated how the transient state performance and robustness of Adaptive RED are influenced by its adaptive control mechanism. In what follows, we therefore apply our analytic results to Adaptive RED, and discuss the

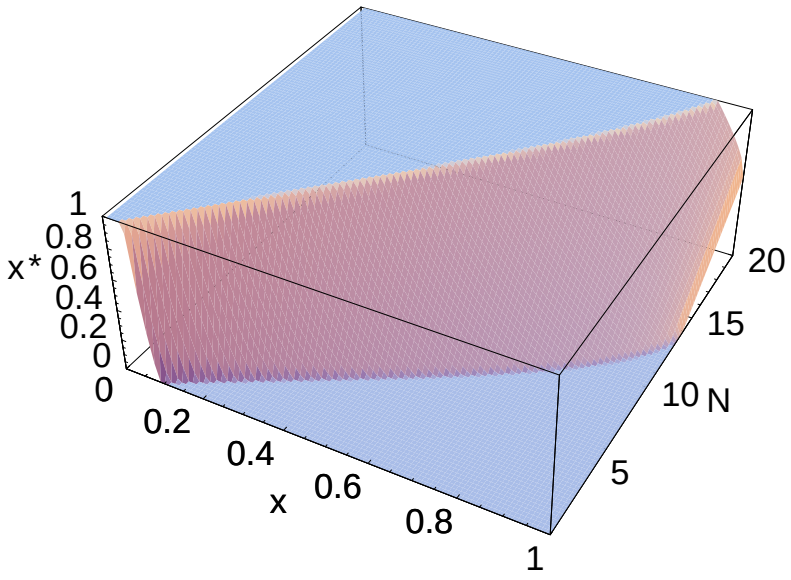

Figure 24: Relation between the number of active TCP connections N and RED queue occupancy with $\mathcal{G}_\phi$ (concave) for ϕ = 1.0.

steady state and transient state performances of Adaptive RED. Although our analysis is not for Adaptive RED but for RED, much insight on Adaptive RED can be gained from our analytic results because of the following reason. The algorithm of Adaptive RED is the same as that of RED except that Adaptive RED dynamically adjusts the maximum packet marking probability max_p according to its average queue length. Hence, at a relatively small time-scale, Adaptive RED can be considered as RED with the roughly optimized maximum packet marking probability max_p. In other words, by analyzing the steady state and transient state performances of RED with different maximum packet marking probabilities, we can investigate the performance of Adaptive RED. As we have explained in Subsection 4.1, the average queue length of RED is dependent on the number of active TCP connections. We can therefore investigate how the performance of Adaptive RED is affected by changing the maximum packet marking probability max_p according to the number of active TCP connections.

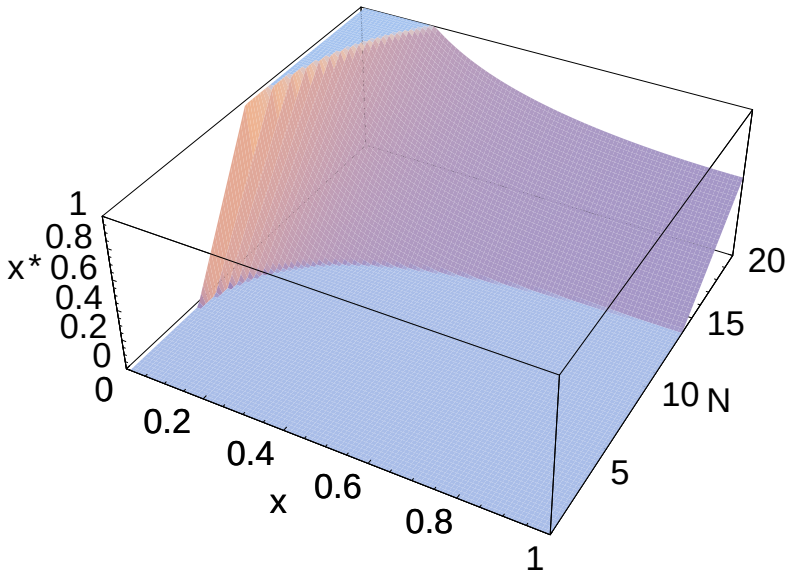

Figure 25: Relation between the number of TCP connections N and RED queue occupancy with $\mathcal{H}_\phi$ (convex) for ϕ = 1.0.

We consider three cases: i.e., the number of active TCP connections N is fixed at 5, 10, or 20. In these cases, by the adaptive control of the maximum packet marking probability max_p, the average queue length of Adaptive RED will be independent of the number of active TCP connections, and will converge to a constant value in steady state. Hence, we assume that the maximum packet marking probability max_p is fixed at 0.02, 0.1, or 0.4 by the control of Adaptive RED when the number of TCP connections N is 5, 10, or 20, respectively. For each case, the relation of the queue occupancy in the x–x^* plane is shown in Figs. 26, 27, or 28, respectively. By comparing these figures, it is evident that the average queue length of Adaptive RED in the steady state (i.e., the intersection with the straight line $x^* = x$) is almost identical. It is also evident that the gradient (dx^*/dx) of Eq. (23) is negligibly affected by the maximum packet marking probability max_p. On the basis of these observation, we find that the adaptive control mechanism of Adaptive RED, which dynamically changes the maximum packet marking probability, improves its steady

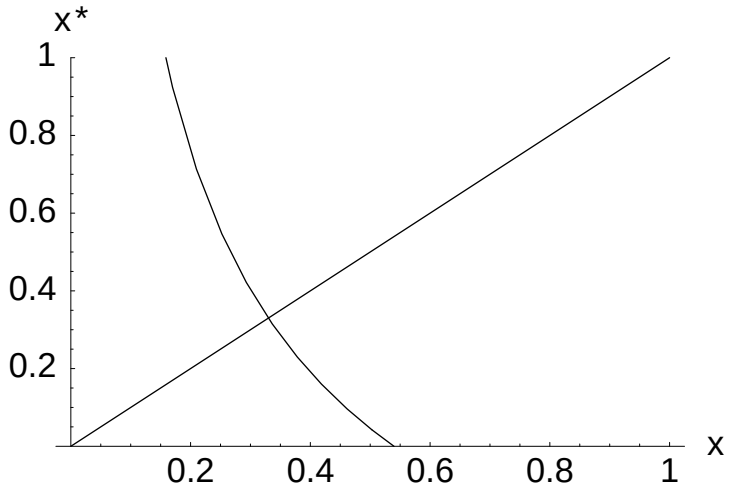

Figure 26: Adaptive RED queue occupancy in the x–x^* plane for $N = 5$ and $max_p = 0.02$.

state behavior, but it does not improve the transient state behavior. However, the performance of Adaptive RED can be improved by using a concave function as the packet marking function.

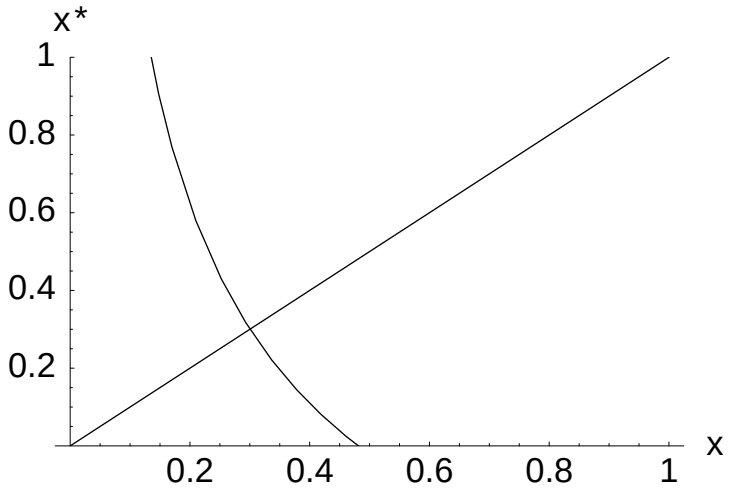

Figure 27: Adaptive RED queue occupancy in the x–x^* plane for $N = 10$ and $max_p = 0.1$.

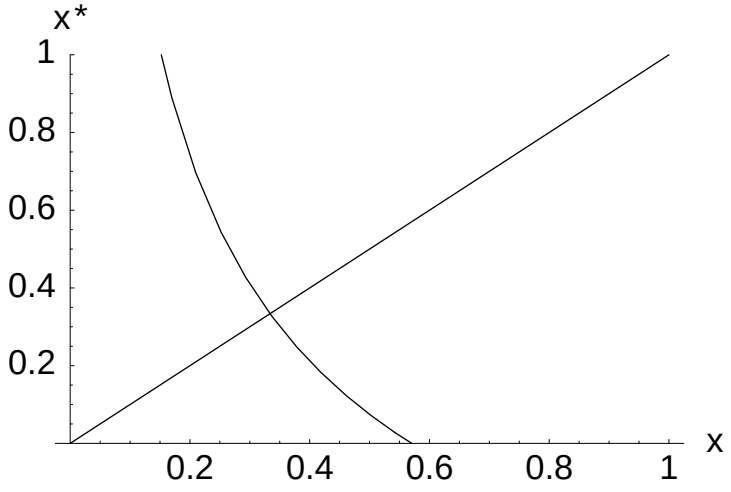

Figure 28: Adaptive RED queue occupancy in the x–x^* plane for $N = 20$ and $max_p = 0.4$.

5 Conclusion

In the first part of 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 [8]. 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 used a control theoretic approach, which is based on the transfer function describing the relation between input and output not in time domain but in frequency domain. Using the transfer function, various characteristics of a feedback system can be investigated. 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.

In the second part of this thesis, we have discussed how the packet loss probability of RED should be determined from its average queue length by utilizing the results of steady state analyses of TCP [8] and RED [16]. By examining several numerical examples, we have investigated the performance of the RED router in three cases — when the function that determines the packet loss probability is either linear, concave, or convex. Consequently, we have found that the transient behavior and the robustness to variation in the number of TCP connections can be improved by using a concave function for determining the packet loss probability of RED. Moreover, we have discussed the characteristics of Adaptive RED with respect to our analysis result. Consequently, we have found that an adaptive control mechanism of Adaptive RED, which dynamically changes the maximum packet loss probability, improves the steady state behavior, but it does not improve the transient state behavior.

Our analytic results have clearly shown that the control of RED, which discards arriving pack-

ets with a probability proportional to its average queue length, has several problems associated with steady state behavior and robustness. In recent years, various AQM mechanisms which solve these drawbacks of RED have been proposed. However, most of these AQM mechanisms use a function which is linear to the average queue length for determining the packet marking probability. Namely, the problems found in this thesis have not been taken into account. As future works, by applying our analytic result obtained in this thesis we therefore intend to design an AQM mechanism by taking account of not only steady state behavior but also transient state behavior and robustness.

Acknowledgements

I would like to express my sincere appreciation to Professor Masayuki Murata of Osaka University, who introduced me to the area of computer networks including the subjects in this thesis. All works of this thesis would not have been possible without the support of Associate Professor Hiroyuki Ohsaki of Osaka University. It gives me great pleasure to acknowledge his assistance. He has been constant sources of encouragement and advice through my studies and preparation of this manuscript. Thanks are also due to Professor Hideo Miyahara and Professor Shinji Shimojo of Osaka University who gave me insightful advises. I am indebted to Associate Professor Masashi Sugano of Osaka Prefecture College of Health Sciences, Associate Professor Ken-ichi Baba of Osaka University, Associate Professor Naoki Wakamiya of Osaka University, Associate Professor Go Hasegawa of Osaka University, Research Assistant Shin'ichi Arakawa of Osaka University and Research Assistant Ichinoshin Maki of Osaka University who gave me helpful comments and feedbacks. I would like to thank many friends and colleagues in the Department of Informatics and Mathematical Science of Osaka University for their valuable suggestions. Finally, I am deeply grateful to my parents who loved and supported me continuously.

References

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

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

[3] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith, "Tuning RED for web traffic," in *Proceedings of ACM SIGCOMM 2000*, pp. 139–150, Aug. 2000.

[4] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A control theoretic analysis of RED," Tech. Rep. TR 00-41, CMPSCI, July 2000.

[5] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proceedings of IEEE INFOCOM 2000*, pp. 1435–1444, Mar. 2000.

[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] H. M. Alazemi, A. Mokhtar, and M. Azizoglu, "Stochastic approach for modeling random early detection gateways in TCP/IP networks," in *Proceedings IEEE International Conference on Communications 2001*, June 2001.

[8] H. Ohsaki and M. Murata, "Steady state analysis of the RED gateway: stability, transient behavior, and parameter setting," *IEICE Transactions on Communications*, vol. E85-B, pp. 107–115, Jan. 2002.

[9] 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.

[10] M. May, J. Bolot, C. Diot, and B. Lyles, “Reasons not to deploy RED,” in *Proceedings of IWQoS ’99*, pp. 260–262, Mar. 1999.

[11] S. Floyd, “Discussions on setting RED parameters,” Nov. 1997. available at `http://www.aciri.org/floyd/REDparameters.txt`.

[12] S. Floyd, “Recommendations on using the gentle variant of RED,” May 2000. available at `http://www.aciri.org/floyd/red/gentle.html`.

[13] J. Aweya, M. Ouellette, D. Y. Montuno, and A. Chapman, “An adaptive buffer management mechanism for improving TCP behavior under heavy load,” in *Proceedings of IEEE International Conference on Communications (ICC 2001)*, pp. 3217–3223, 2001.

[14] T. J. Ott, T. V. Lakshman, and L. Wong, “SRED: Stabilized RED,” in *Proceedings of IEEE INFOCOM ’99*, pp. 1346–1355, Mar. 1999.

[15] W.-C. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, “Techniques for eliminating packet loss in congested TCP/IP networks,” Tech. Rep. CSE-TR-349-97, Oct. 1997.

[16] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP throughput: a simple model and its empirical validation,” in *Proceedings of ACM SIGCOMM ’98*, pp. 303–314, Sept. 1998.

[17] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[18] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, “On designing improved controllers for AQM routers supporting TCP flows,” in *Proceedings of IEEE INFOCOM 2001*, pp. 1726–1734, 2001.

[19] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: an algorithm for increasing the robustness of RED," available at `http://www.icir.org/floyd/papers/adaptiveRed.pdf`, Aug. 2001.

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

[21] M. Kisimoto, H. Ohsaki, and M. Murata, "Analyzing the impact of TCP connections variation on transient behavior of RED gateway," in *Proceedings of the 16th International Conference on Information Networking (ICOIN-16)*, pp. 10A2.1–10A2.12, Jan. 2002.

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

[23] N. S. Nise, *Control Systems Engineering with MATLAB*. John Wiley & Sons, 3 ed., Feb. 2000.