Stability Analysis of a Window-based Flow Control Mechanism for TCP Connections with Different Propagation Delays

Keiichi Takagaki Hiroyuki Ohsaki Masayuki Murata

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

(Phone) +81-6-6850-6616
(Fax) +81-6-6850-6589
(E-mail) takagaki@ics.es.osaka-u.ac.jp

Abstract

A feedback-based congestion control mechanism is essential to realize an efficient data transfer service in packet-switched networks. TCP (Transmission Control Protocol), a sort of feedback-based congestion control mechanism, has been widely used in the current Internet. Recently-proposed TCP Vegas is another version of TCP mechanism, and achieves much better performance than current TCP Reno. In this paper, we focus on a window-based flow control mechanism based on the congestion avoidance mechanism of TCP Vegas, and analyze its stability using a control theoretic approach. The main objective of this paper is to analyze the dynamics of the window-based flow control mechanism when TCP connections have different propagation delays. Through the analysis, we show that the system can be stabilized by choosing the control parameter of each connection proportionally to its round-trip propagation delay.

1 Introduction

A feedback-based congestion control mechanism is essential to realize an efficient data transfer services in packed-switched networks. TCP (Transmission Control Protocol), a sort of feedback-based congestion control mechanisms, has been widely used in the current Internet. In this paper, we are devoted to studying the functionality of a congestion control mechanism of TCP, which controls a congestion level of the network by regulating a window size of a source host according to feedback information obtained from the network (via the receiver host).

Recently, another version of TCP called TCP Vegas has been proposed by Brakmo *et al.*, which can achieve better performance than TCP Reno [1]. TCP Vegas has following advantages over existing TCP Reno: (1) a new time-out mechanism, (2) an improved congestion avoidance mechanism, and (3) a modified slow-start mechanism. In particular, the congestion avoidance mechanism of TCP Vegas controls the number of on-the-fly packets in the network. More specifically, TCP Vegas measures an RTT (Round Trip Time), which is elapsed time from a packet transmission to the receipt of its corresponding ACK (ACKnowledgment) packet. It then uses the measured RTT as feedback information from the network. Namely, if the measured RTT is getting large, the source host of TCP Vegas conjectures that the network is falling into congestion. Then, the window size is throttled. If measured RTTs become small, on the other hand, the source host recognizes that the network is relieved from the congestion, and increases the window size again. In TCP Vegas, it is not necessary for the source host to wait a packet loss in the network to detect congestion. This is an advantage of TCP Vegas over other versions of TCP. With this mechanism, the window size of a source host is expected to converge to a constant value in steady state. The simulation and experimental results show that the congestion control mechanism of TCP Vegas leads to 37–71 % higher throughput than that of TCP Reno [1].

Dynamics of TCP Vegas has been analytically investigated by several researchers [2-4]. In those papers, the evolution of a window size has been approximated by a fluid model, and the throughput of each connection has been obtained. However, the analytic model used in those papers has focused only on a single connection [2, 3], or two connections [4]. Therefore, those results are not applicable to a real network. In addition, in the above papers, stability of TCP Vegas has not been investigated at all. Since TCP Vegas is essentially a feedback congestion control, a stable operation of the control mechanism is very important, but the approach based on the fluid model cannot examine such an aspect.

One exception can be found in [5], where the authors explicitly derive the stability condition and optimal setting of control parameters by applying the control theory. Control parameters for best transient performance can also be investigated by their approach. Further, the analytic model considered in [5] allows multiple TCP connections. However, the authors assume a single bottleneck link, and a more important shortcoming is an assumption on the propagation delays; all connections have identical propagation delays. Since in the real network, each connection usually has a different propagation delay, and the difference of feedback delays must affect the stability and performance of TCP connections.

As a next step for making the control theoretic approach to be useful for the network using the feedback–based congestion control (i.e., the Internet), we consider a network where each connection is allowed to have a different propagation delay by extending a previous work [5]. First, we derive fixed points of the window size and the number of the packets in the router's buffer in steady state. Based on these results, we then derive throughput of each connection in steady state, and show that the throughput of each connection become fair regardless of its propagation delay if we set a control parameter that controls the number of on-the-fly packets equally. We also derive the stability condition of the window-based flow control mechanism. Then, we quantitatively show how the stability region of control parameters is affected by network parameters such as the processing speed of the router and the propagation delay.

Analyses of feedback-based congestion control mechanisms using control theoretic approaches can be found in the literature. In particular, several papers including [6-8] have analyzed a feedback-based congestion control mechanism for the network model where each connection has a different propagation delay. The authors of these papers have focused on a rate-based congestion control mechanism in ATM networks, and have designed a rate controller based on the optimal control theory. However, their approaches are not applicable to the window-based flow control mechanism since the behaviors of a rate-based and a window-based congestion control mechanisms are essentially different.

Organization of this paper is as follows. In Section 2, we describe the window-based flow control mechanism based on the congestion avoidance mechanism of TCP Vegas, followed by introduction of its analytic model. In Section 3, stability analysis of the window-based flow control mechanism is performed by applying control theory. In Section 4, the effect of control parameters on system stability is investigated by illustrating several numerical examples. The numerical results presented in Section 4 are validated by using simulation experiments in Section 5. In Section 6, we conclude this paper and discuss future works.

2 Analytic Model

The analytic model used in this paper is illustrated in Fig. 1. In this figure, several TCP connections are established through a single bottleneck router. There are M groups of connections where connections in each group are assumed to have an identical propagation delay. Let τ_m be the propagation delay of connections in group m $(1 \leq m \leq M)$. We assume $\tau_1 < \tau_2 < \cdots < \tau_{M-1} < \tau_M$ without loss of generality. We introduce an irreducible positive integer Δ_m $(1 \leq m \leq M)$ as the ratio of propagation delay τ_m, such that

$$\frac{\tau_1}{\Delta_1} = \frac{\tau_2}{\Delta_2} = \cdots \frac{\tau_{M-1}}{\Delta_{M-1}} = \frac{\tau_M}{\Delta_M}.$$

By assuming that the waiting time of a packet at the router's buffer is negligible, the ratio of RTTs for connections in group m is given by Δ_m. In TCP Vegas, a source host changes its window size once per RTT [1]. Therefore, the system can be represented by a discrete-time model where its time slot is τ_m/Δ_m. In other words, connections in group m change their window sizes every Δ_m slots.

Let N_m be the number of connections in group m, and $w_{m,n}(k)$ be the window size of the source host n

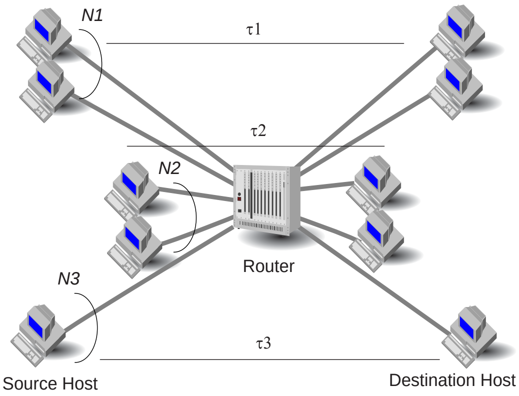

Figure 1: Analytic model for $M = 3$.

$(1 \leq n \leq N_m)$ in group m at slot k. Namely, the source host n in group m is allowed to send $w_{m,n}(k)$ packets during its RTT. We assume that all source hosts always have enough data to transmit so that every connection always sends the number $w_{m,n}(k)$ of packets during its RTT. Further, let $q(k)$ be the number of packets in the router's buffer at slot k, and L be the capacity of the router's buffer. The router is assumed to process incoming packets according to FIFO (First-In First-Out) discipline. The processing speed of the router is denoted by B. By assuming the packet size to be fixed, we use the unit of "packet" for the window size $w_{m,n}(k)$ and the capacity of the router's buffer L, and "packet/ms" for the processing speed of the router B.

Since the window-based flow control mechanism allows the source host to emit the number $w_{m,n}(k)$ of packets per RTT, a connection in group m sends the number $w_{m,n}/\Delta_m$ of packets per slot on the average. Therefore, the number of packets in the router's buffer at slot $k+1$ is given by the following equation

$$q(k+1) = \min\left[\max\left(\sum_{m=1}^{M}\sum_{n=1}^{N_m}\frac{w_{m,n}(k)}{\Delta_m} - B\,\Delta, 0\right), L\right],$$

where Δ is the length of a single slot.

In TCP Vegas, the source host measures its RTT and changes its window size based on the observed RTT. More specifically, source host n in group m calculates

$$d_{m,n}(k) = \left(\frac{w_{m,n}(k)}{\tau_m} - \frac{w_{m,n}(k)}{r_m(k)}\right) \times \tau_m, \tag{1}$$

from its measured RTT $r_m(k)$. Note that $r_m(k)$ is dependent on the number of packets in the router's buffer, and given by the sum of the propagation delay and the waiting time at the router's buffer. That is,

$$r_m(k) = \tau_m + \frac{q(k)}{B}.$$

The source host changes its window size once every RTT according to $d_{m,n}(k)$. Namely, the window size $w_{m,n}(k + \Delta_m)$ is changed as

$$w_{m,n}(k + \Delta_m) = \begin{cases} w_{m,n}(k) + 1, & \text{if } d_{m,n}(k) < \alpha_{m,n} \\ w_{m,n}(k) - 1, & \text{if } d_{m,n}(k) > \beta_{m,n} \\ w_{m,n}(k), & \text{otherwise} \end{cases}, \tag{2}$$

where $\alpha_{m,n}$ and $\beta_{m,n}$ are control parameters that determine the number of on-the-fly packets in the router's buffer. In this paper, we modify Eq. (2) as

$$w_{m,n}(k + \Delta_m) = \max(w_{m,n}(k) + \delta_{m,n}(\gamma_{m,n} - d_{m,n}(k)), 0), \tag{3}$$

where $\delta_{m,n}$ is a control parameter that determines the amount of the window size change. The purpose of introducing $\delta_{m,n}$ is not only for enabling application of a control theory, but also for improving transient performance [9]. In [10], it has been reported that fairness among connections cannot be satisfied when $d(k)$ lies in [$\alpha_{m,n}$, $\beta_{m,n}$]. In our analytic model, we therefore unify both $\alpha_{m,n}$ and $\beta_{m,n}$ in Eq. (2) into a single one, $\gamma_{m,n}$, as in Eq. (3). With this modification, fairness among connections can be improved [10]. Intuitively, γ controls the number of on-the-fly packets in the network for each connection.

3 Stability Analysis

In what follows, we assume that initial values of window sizes of all source hosts are identical, and also assume that control parameters of connections in the same group are identical. For brevity, the control parameter of source hosts in group m is represented by $\delta_m (\equiv \delta_{m,n})$ $(1 \leq m \leq M, 1 \leq n \leq N_m)$. Provided that all source hosts change their window sizes according to Eq. (3), the number of packets in the router's buffer at slot $k + 1$ is given by

$$q(k+1) = \min\left(\max\left(\sum_{m=1}^{M} \frac{N_m w_m(k)}{\Delta_m} - B\,\Delta, 0\right), L\right), \tag{4}$$

where $w_m(k) \equiv w_{m,n}(k)$ $(1 \leq n \leq N_m)$.

Let w_m^*, q^* and d_m^* be the fixed points of $w_m(k)$, $q(k)$ and $d_m(k)$ in steady state, respectively. By using Eqs. (1), (3) and (4), w_m^*, q^* and d_m^* can be obtained easily. Let $\mathbf{x}(k)$ be the difference of the system state from its fixed points at slot k, i.e.,

$$\mathbf{x}(k) \equiv \begin{bmatrix} w_1(k) & - & w_1^* \\ & \vdots & \\ w_M(k) & - & w_M^* \\ q(k) & - & q^* \end{bmatrix}.$$

Since $w_m(k)$ is a non-linear equation, we linearize it around the fixed point. By letting Δ_L be the LCM (Lowest Common Multiple) of Δ_1, $\Delta_2 \cdots \Delta_{M-1}$, Δ_M, $\mathbf{x}(k + \Delta_L)$ can be written as

$$\mathbf{x}(k + \Delta_L) = \mathbf{A}\, \mathbf{x}(k), \tag{5}$$

where $\mathbf{A}$ is a state transition matrix. Stability and transient behavior of the system around the fixed point is determined by eigenvalues of the matrix $\mathbf{A}$. More specifically, the fixed point is locally asymptotically stable if and only if all roots $s_i(1 \leq i \leq M + 1)$ of the characteristic equation $D(s) = |sI - A| = 0$ lie in the unit circle [11]. It can be easily checked by using the Jury's criterion if the matrix $\mathbf{A}$ satisfies this condition or not [11].

In the following, we discuss the case of $M = 2$ (i.e., two groups of connections) for $\Delta_1 = 1$ and $\Delta_2 = 2$ (i.e., the ratio of the propagation delays is 1:2) due to space limitation. For the cases of $M > 3$ or other ratios of propagation delays, the same approach can be easily applied.

In the case of $M = 2$, $\Delta_1 = 1$, and $\Delta_2 = 2$, the fixed points of the system are given by

$$w_1^* = \frac{\gamma_1(v_1 + v_2 + v_3)}{2v_1} \tag{6}$$

$$w_2^* = \frac{\gamma_2(v_2 + v_3)}{v_1} \tag{7}$$

$$q^* = \frac{v_1 - v_2 + v_3}{4}, \tag{8}$$

where

$$\begin{aligned} v_1 &= 2(N_1\gamma_1 + N_2\gamma_2) \\ v_2 &= 2B\tau_1 + N_2\gamma_2 \\ v_3 &= \sqrt{(v_1 - v_2)^2 + 8B\tau_1 v_1}. \end{aligned}$$

When the waiting time at the router's buffer is very small compared to propagation delays, w_m^* and q^* can be approximated as

$$\begin{aligned} w_m^* &\simeq \frac{\gamma_m B\,\tau_m}{N_1\gamma_1 + N_2\gamma_2} && (9)\\ q^* &\simeq N_1\gamma_1 + N_2\gamma_2. && (10)\end{aligned}$$

Equation (9) indicates that the fixed point of the window size w_m^* is proportional to the processing speed of the router B and the propagation delay τ_m, and is inversely proportional to the number of connections N_1 and N_2. Equation (10) indicates that the fixed point of the number of the packets in the router's buffer q^* is the sum of γ_m's of all source hosts. Thus, the number of packets in the router's buffer in steady state can be controlled by appropriately choosing a control parameter γ_m at each source host.

We then derive the throughput of each connection. The throughput ρ_m of connections in group m is given by

$$\rho_m \equiv \frac{w_m^*}{r_m} = \frac{w_m^*}{\tau_m + \frac{q^*}{B}}.$$

From Eq. (1), we find that d_m^* converges to γ_m in steady state. Using Eq. (3) gives

$$w_m^* = \frac{\gamma_m(\tau_m + \frac{q^*}{B})}{\frac{q^*}{B}}.$$

Hence, the throughput ρ_m is obtained as

$$\rho_m = \frac{\gamma_m B}{q^*}. \qquad (11)$$

The above equation suggests that the ratio of throughput ρ_1/ρ_2 of connections in groups 1 and 2 is simply determined by the ratio of γ_1 and γ_2. It means that the ratio of throughput of connections is dependent only on the control parameter γ_m, and is independent of other parameters; the propagation delay τ_m and the number of connections N_m. If we choose the control parameter appropriately, the router would be fully utilized. In this case, ρ_m is given by a simple equation.

$$\rho_m = \frac{\gamma_m B}{N_1\gamma_1 + N_2\gamma_2}$$

Therefore, if γ_m's of all connections are set equally, the throughput of all connections become identical, leading to a fair bandwidth allocation among all connections. However, this sort of fairness can be achieved only when there exists a single bottleneck router in the network. We have shown that the throughput of TCP Vegas is determined not only by a control parameter but also by the number of routers on the path [12]. Therefore, in real networks, it is insufficient to set γ_m equally to all connections for achieving a fair bandwidth allocation to connections. For example, γ_m should be chosen according to the number of bottleneck router's on the path. However, parameter tuning of γ_m for such network configuration is beyond the scope of this paper.

In the case of $M = 2$, $\Delta_1 = 1$, and $\Delta_2 = 2$, the matrix $\mathbf{A}$ becomes

$$\mathbf{A} \;=\; \begin{bmatrix} a_1^2 + N_1 b_1 & N_2 b_1/2 & a_1 b_1 \\ 0 & a_2 & b_2 \\ N_1 a_1 & N_2/2 & N_1 b_1 \end{bmatrix}, \tag{12}$$

where a_m and b_m (m=1, 2) are defined as

$$\begin{aligned} a_m &= 1 - \delta_m + \frac{B\delta_m\tau_m}{B\tau_m + q^*}, \\ b_m &= -\frac{B\delta_m\tau_m w_m^*}{(B\tau_m + q^*)^2}. \end{aligned}$$

4 Numerical Examples

In this section, we present several numerical examples and discuss how the stability region is affected by a choice of control parameters and system parameters. Figure 2 shows boundary lines of the stability region on δ_1–δ_2 plane for several values of processing speed of the router B. In this figure, the following parameters are used: the number of connections N_1 = N_2 = 10, the round-trip propagation delay τ_1 = 1 [ms] and τ_2 = 2 [ms], and the control parameter γ_1 = γ_2 = 3 [packet]. The processing speed of the router B is changed from 2 to 2,000 [packet/ms]. This figure means that the system becomes stable when the point (δ_1, δ_2) lies inside the boundary line. That is, for the window-based congestion control mechanism to be stable, we should choose the point (δ_1, δ_2) in the region surrounded by the boundary line and both x- and y- axes.

One can find from the figure that the stability region heavily depends on the processing speed of the router B. One can also find that the maximum value of δ_2 for connections with a large propagation delay is larger than the maximum value of δ_1 for connections with a small propagation delay. This tendency becomes more noticeable as the processing speed of the router B becomes large. For example, when $B = 2,000$ [packet/ms], the maximum values of δ_1 and δ_2 for stable operation are 2 and 4, respectively. Note that the ratio of δ_1 and δ_2

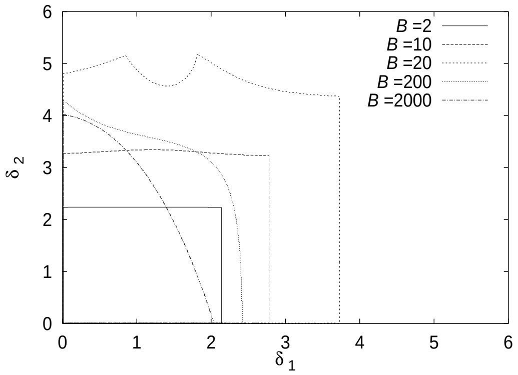

Figure 2: Stability region for different processing speeds of the router B ($N_1 = N_2 = 10$, $\tau_1 = 1$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet])

is equal to the ratio of their propagation delays, τ_1 and τ_2. This can be explained as follows. In TCP Vegas, a source host changes its window size once per RTT. Hence, a connection with a larger propagation delay changes its window size less frequently so that it has less influence on system stability. In other words, if a connection with a small propagation delay changes its window size excessively, the system is likely to become unstable. So if δ_m's of all connections are equal, stability of the system is mostly determined by the connection with the smallest propagation delay.

Figure 2 also indicates that when the processing speed of the router B is small, the maximum value of δ_m is almost independent of the propagation delay. For example, when $B = 2$ [packet/ms], the system becomes stable if both δ_1 and δ_2 are set less than 2. This is because when the processing speed of the router B is very small, the waiting time at the router's buffer is much larger than the propagation delay. Namely, the observed RTT of a connection is not so affected by its propagation delay since the waiting time at the router's buffer is the dominant part of its RTT. Consequently, frequency of the window size change becomes almost same in all connections so that the maximum value of δ_m becomes identical.

Figure 3 shows the stability region for different values of propagation delays τ_1 from 0.1 [ms] to 100 [ms]. In this figure, values of control parameters and system parameters are equal to those in Fig. 2, whereas the processing speed of the router B is fixed at 20 [packet/ms]. In this figure, the ratio of propagation delays is fixed at 1:2. Namely, τ_2 is always twice of τ_1. By comparing Figs. 2 and 3, one can find that boundary lines in these figures are almost identical. Such a correspondence can be easily explained from Eqs. (6)-(8) and (12).

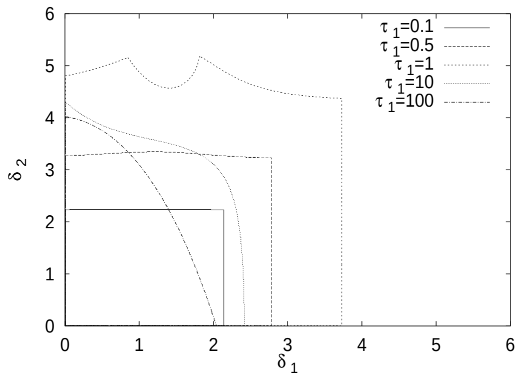

Figure 3: Stability region for different propagation delays τ_1 ($B = 20$ [packet/ms], $N_1 = N_2 = 10$, $\gamma_1 = \gamma_2 = 3$ [packet])

Namely, all B's and τ_m's in these equations take the product form of $B \times \tau_m$. This suggests an interesting fact that the same effect is obtained by increasing the processing speed of the router B and by increasing the propagation delay τ_m in the window-based congestion control mechanism. Such a characteristic of the window-based flow control mechanism indicates that the bandwidth–delay product, $B \times \tau_m$, is one of key factors that determine system stability. Intuitively, the bandwidth–delay product, $B \times \tau$, represents the number of on-the-fly packets on all transmission links. Our analytic results suggests that the system becomes less stable as the number of on-the-fly packets increases.

We next show the stability region for different numbers of connections N_1 and N_2 in Figs. 4 and 5, respectively. In these figures, the following parameters are used: the processing speed of the router $B = 20$ [packet/ms], the propagation delay $\tau_1 = 1$ [ms], $\tau_2 = 2$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet]. In Fig. 4, the number of connections N_2 is fixed at 10 but N_1 is changed from 1 to 1,000. On the contrary, in Fig. 5, N_1 is fixed at 10 but N_2 is changed from 1 to 1,000. These figures show that the maximum value of δ_m for stable operation becomes small as the number of connections becomes large. For example, one can find that the maximum value of δ_1 becomes small as N_1 becomes large in Fig. 4. This phenomenon can be explained as follows. Since RTTs of connections in a group are identical, these connections change their window sizes synchronously. When the number of connections in a group increases, the amount of the window size change becomes large. Therefore, the system tends to become less stable so that δ_m should be small for avoiding unstable operation.

However, Figs. 4 and 5 also show that when $\delta_1 < 2$ and $\delta_2 < 2$, the system is always stabilized regardless

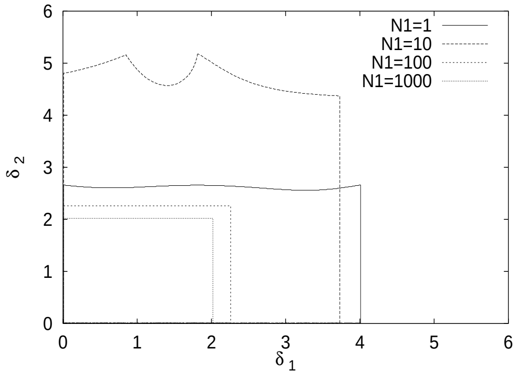

Figure 4: Stability region for different numbers of connections N_1 ($B = 20$ [packet/ms], $N_2 = 10$, $\tau_1 = \tau_2 = 1$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet])

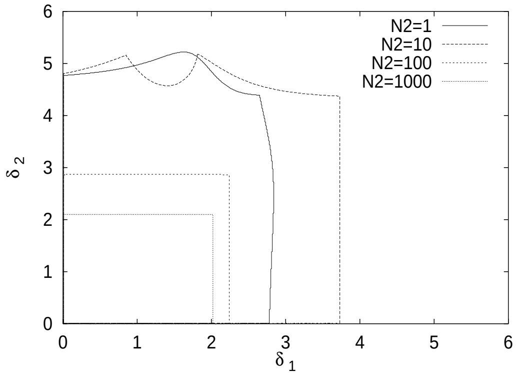

Figure 5: Stability region for different numbers of connections N_2 ($B = 20$ [packet/ms], $N_1 = 10$, $\tau_1 = \tau_2 = 1$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet])

of the number of connections. Since the number of active connections in a real network changes frequently, and since it is difficult to prospect, setting of control parameters of $\delta_1 < 2$ and $\delta_2 < 2$ would be desired for practical purposes. Moreover, it is expected from Figs. 2 through 5 that there exists a region where the system can always be stabilized for any network parameters. By increasing the processing speed of the router B, the number of connections N_m, the propagation delay τ_m or the control parameter γ_m to infinity, the stability region for any network parameters can be easily found: $\delta_1 < 1$ and $\delta_2 < 2$. However, to select the optimal point of (δ_1, δ_2), we should consider transient performance in addition to system stability. The optimal (δ_1, δ_2) that leads to the best transient behavior can be calculated by the same method used in [5]. However, the consideration about setting optimal (δ_1, δ_2) for the best transient behavior is our future work.

Finally, we show the dynamics of the window-based flow control mechanism in stable and unstable cases. We choose the processing speed of the router $B = 20$ [packet/ms], and (δ_1, δ_2) = (3.0, 4.0) in Fig. 6 (stable case) or (δ_1, δ_2) = (4.0, 5.0) in Fig. 7 (unstable case). The other parameters are identical to those used in Fig. 2. In these figures, the evolutions of the window size $w_m(k)$ and the number of the packets at the router's buffer $q(k)$ are plotted. These figures are obtained by numerically computing the system state using Eq. (5). The initial values of the window size $w_m(k)$ and the number of the packets at the router's buffer $q(k)$ are set to 80 % of the fixed points, w^* and q^*.

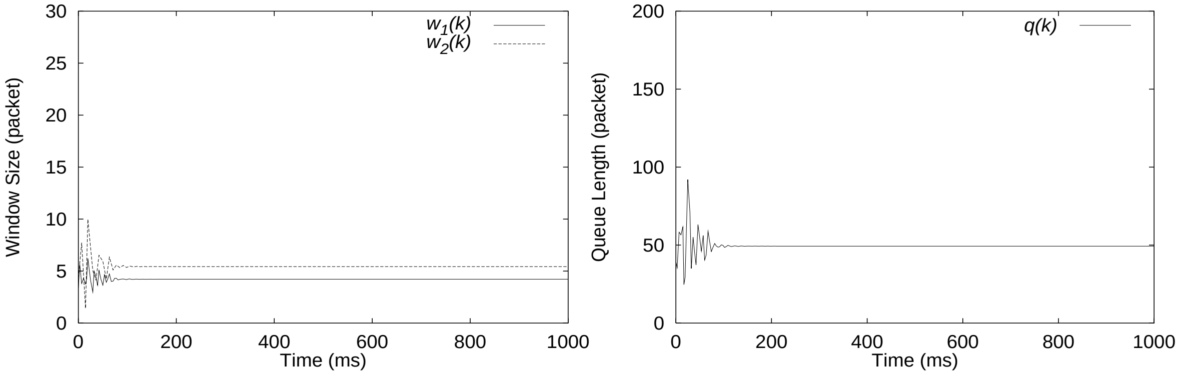

Figure 6: Stable behavior ($B = 20$ [packet/ms], $N_1 = N_2 = 10$, $\tau_1 = 1$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet], $\delta_1 = 3.0, \delta_2 = 4.0$)

In the stable case (Fig. 6), the system becomes stable in 150 [ms] as expected. However, in the unstable case (Fig. 7), both the window size and the number of the packets at the router's buffer oscillate indefinitely.

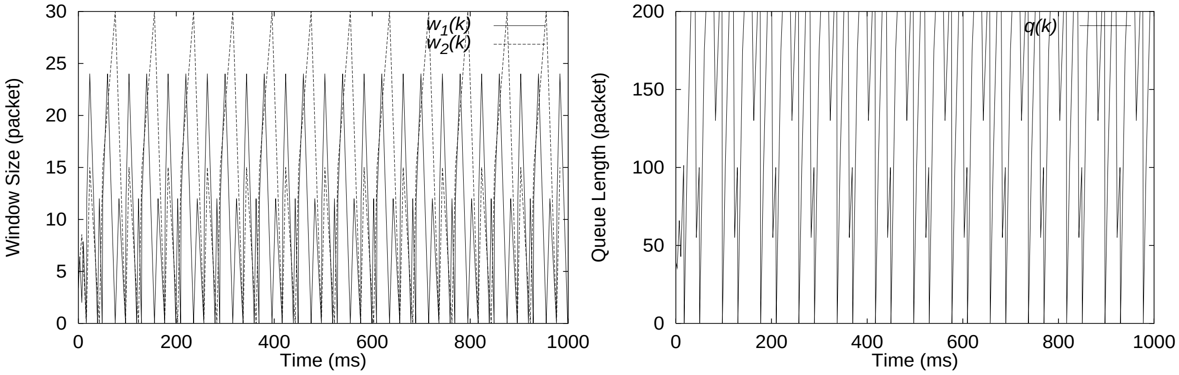

Figure 7: Unstable behavior ($B = 20$ [packet/ms], $N_1 = N_2 = 10$, $\tau_1 = 1$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet], $\delta_1 = 4.0, \delta_2 = 5.0$)

5 Simulation Results

In this section, several simulation results are shown for validating our analysis presented in Section 3. We use the following parameters: the processing speed of the router $B = 20$ [packet/ms], the number of connections $N_1 = N_2 = 1$, the propagation delay $\tau_1 = 1$ [ms], $\tau_2 = 2$ [ms], the control parameter $\gamma_1 = \gamma_2 = 3$ [packet]. For the control parameter δ_m, we use (δ_1, δ_2) = (1.0, 3.0) as a stable case and (3.0, 1.0) as unstable case. Figure 8 shows the stability region for these parameters obtained from our analysis. Dynamics of the windows size $w_m(k)$ and the number of packets at the router $q(k)$ are plotted in Figs. 9 (stable case) and 10 (unstable case), respectively.

Using Eqs.(6)–(8), the fixed points, w_m^* and q^*, in this parameter setting should be

$$\begin{aligned} w_1^* &= 13.6 \\ w_2^* &= 24.2 \\ q^* &= 5.7. \end{aligned}$$

From Figs. 9 and 10, the fixed points in simulation experiments are

$$\begin{aligned} w_1^* &= 15.5 \\ w_2^* &= 22.0 \\ q^* &= 5.3, \end{aligned}$$

which shows close values to analytic ones. The difference between analytic and simulation results is possibly caused by our assumptions that the window size change is synchronous and the waiting time at the router's buffer is neglected.

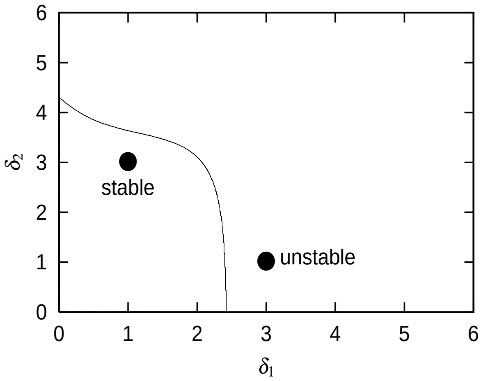

Figure 8: Stability region ($B = 20$ [packet/ms], $N_1 = N_2 = 1$, $\tau_1 = 1$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet])

By comparing Figs. 9 (stable case) and 10 (unstable case), it can be found that both the window size $w_m(k)$ and the number of packets at the router $q(k)$ oscillate excessively when (δ_1, δ_2) is out of the stability region. On the contrary, when (δ_1, δ_2) satisfies the stability condition, the dynamics of the system becomes almost stable although there is slight oscillation in the window size $w_m(k)$ and $q(k)$.

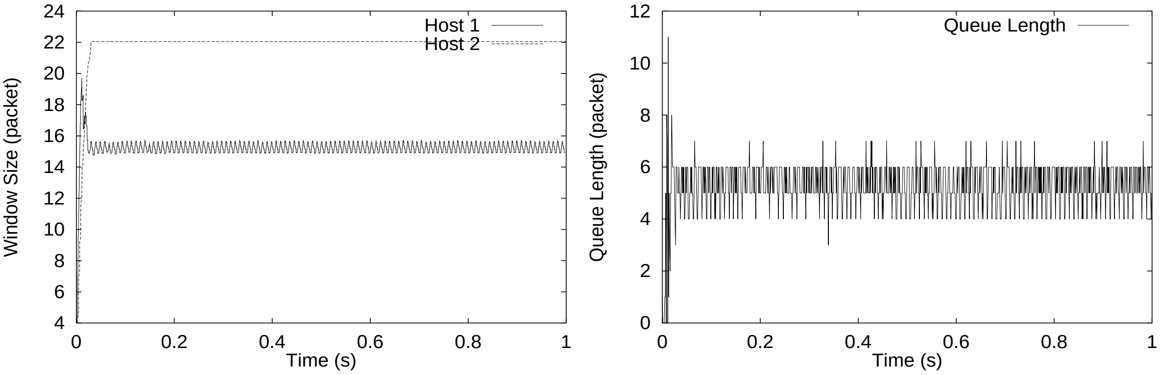

Figure 9: Stable behavior ($B = 20$ [packet/ms], $N_1 = N_2 = 1$, $\tau_1 = 1$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet], $\delta_1 = 1.0$, $\delta_2 = 3.0$)

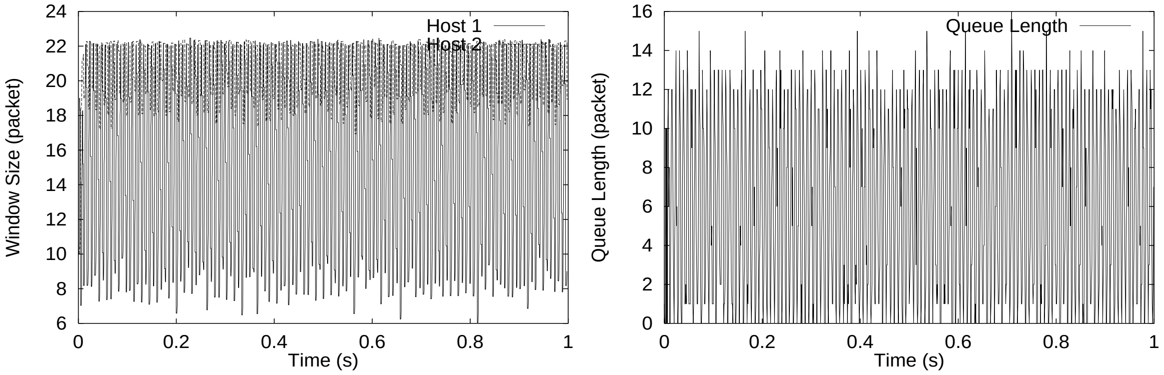

Figure 10: Unstable behavior ($B = 20$ [packet/ms], $N_1 = N_2 = 1$, $\tau_1 = 1$ [ms], $\gamma_1 = \gamma_2 = 3$ [packet], $\delta_1 = 3.0$, $\delta_2 = 1.0$)

6 Conclusion and Future Work

In this paper, we have focused on the window-based flow control mechanism based on the congestion avoidance mechanism of TCP Vegas. We have analyzed its behavior in steady state by applying a control theory. We have considered a network model consisting of TCP connections with different propagation delays. We have first derived the fixed points of the window size and the number of packets in the bottleneck router's buffer. We have shown that a fair bandwidth allocation to all connections can be realized by setting γ_m's of all connections identically regardless of the difference of their propagation delays. We have also derived the stability condition of the window-based flow control mechanism, and have investigated the relation between system stability and network parameters through several numerical examples. We have found that the system can be stabilized by choosing the control parameter of each connection proportionally to its round-trip propagation delay. Simulation results have also been presented for validating our analysis.

Our future work is to find the optimal setting of control parameters for achieving reasonable transient performance as well as stable operation of the system. Moreover, dynamics of the window-based flow control mechanism when a new connection is established should also be investigated. Our ongoing research is to analyze more generic network configurations where there exists more than two bottleneck routers in the network.

References

[1] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proceedings of ACM SIGCOMM '94*, pp. 24–35, October 1994.

[2] O. A. Hellal and E. Altman, "Analysis of TCP Vegas and TCP Reno," in *Proceedings of IEEE ICC '97*, pp. 495–499, June 1997.

[3] A. Kumar, "Comperative performance analysis of versions of TCP in a local network with a lossy link," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 485–498, August 1998.

[4] J. Mo, R. J. La, V. Anantharam, and J. Walrand, "Analysis and comparison of TCP Reno and TCP Vegas," in *Proceedings of IEEE INFOCOM '99*, March 1999.

[5] H. Ohsaki, M. Murata, T. Ushio, and H. Miyahara, "Stability analysis of window-based flow control mechanism in TCP/IP networks," *1999 IEEE International Conference on Control Applications*, pp. 1603–1606, August 1999.

[6] B.-K. Kim and C. Thompson, "Optimal feedback control of ABR traffic in ATM networks," in *Proceedings of IEEE GLOBECOM '98*, pp. 844–848, 1998.

[7] H. Zhang and O. W. Yang, "Design of robust congestion controllers for ATM networks," in *Proceedings of IEEE INFOCOM '97*, pp. 302–309, April 1997.

[8] A. Kolarov and G. Ramamurthy, "A control theoretic approach to the design of closed-loop rate based flow control for high speed ATM networks," in *Proceedings of IEEE INFOCOM '97*, pp. 293–301, April 1997.

[9] H. Ohsaki, M. Murata, T. Ushio, and H. Miyahara, "A control theoretical analysis of a window-based flow control mechanism in TCP/IP networks," submitted to *IEEE Journal on Selected Areas in Communications QoS in the Internet*, 1999.

[10] G. Hasegawa, M. Murata, and H. Miyahara, "Fairness and stability of congestion control mechanism of TCP," in *Proceedings of 11th ITC Special Seminar*, pp. 255–262, October 1998.

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

[12] K. Takagaki, H. Ohsaki, and M. Murata, "Analysis of window-based flow control mechanism in heterogeneous network environment," in preparation, 2000.