A control theoretical analysis of a window-based flow control mechanism for TCP connections with different propagation delays

Hiroyuki Ohsaki[†], Keiichi Takagaki[‡], and Masayuki Murata[†]

[†] Cybermedia Center, Osaka University, Japan
[‡] Graduate School of Engineering Science, Osaka University, Japan

ABSTRACT

A feedback-based congestion control mechanism is essential to realize an efficient best-effort service in high-speed networks. A window-based flow control mechanism called TCP (Transmission Control Protocol), which is 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 can achieve better performance than the current TCP Reno. In our previous works, we have analyzed stability of a window-based flow control mechanism based on TCP Vegas in both homogeneous and heterogeneous networks. In this paper, using our analytic results, we investigate how the dynamics of the window-based flow control mechanism is affected by the difference in propagation delays of TCP connections. We also investigate the effect of various system parameters on transient performance of the window-based flow control mechanism.

Keywords: Window-based flow control mechanism, TCP Vegas, Control theory, Stability, Transient performance

1. INTRODUCTION

A feedback-based congestion control mechanism is essential to realize efficient data transfer services in packet-switched networks. TCP (Transmission Control Protocol) is a feedback-based congestion control mechanism, and has been widely used in the current Internet. For example, a version of TCP mechanism called *TCP Reno* uses packet losses in the network as feedback information since a packet loss implies congestion occurrence in the network.[1,2] In short, the congestion control mechanism of TCP Reno first increases its window size, and as soon as it detects packet losses in the network, it reduces its window size. TCP Reno repeats this process indefinitely.

In 1994, another version of TCP called *TCP Vegas* has been proposed by Brakmo *et al.*, which can achieve better performance than TCP Reno.[3] TCP Vegas has the following advantages over TCP Reno: (1) a new timeout mechanism to detect packet losses much sooner than TCP Reno, (2) an improved congestion avoidance mechanism to control the number of in-flight packets within the network, and (3) a modified slow-start mechanism to prevent from sending packets at an excessively high rate. In particular, the heart of TCP Vegas is its congestion avoidance mechanism; TCP Vegas measures an RTT (Round-Trip Time), which is time elapsed from a packet transmission to the receipt of its corresponding ACK (ACKnowledgment) packet. TCP Vegas uses the measured RTT as a congestion indication from the network. It has been reported that the congestion control mechanism of TCP Vegas leads to 37–71 % higher throughput than that of TCP Reno.[3]

In the literature, there have been several simulation and experimental studies on TCP Vegas.[4–6] Also there have been analytic studies of TCP Vegas.[7–10] In those analytical studies, the evolution of a window size is approximated by a fluid model, and the throughput of a TCP connection is derived. However, those analytical studies use a

Further author information: (Send correspondence to Hiroyuki Ohsaki)
Hiroyuki Ohsaki: oosaki@cmc.osaka-u.ac.jp
Keiichi Takagaki: takagaki@ics.es.osaka-u.ac.jp
Masayuki Murata: murata@cmc.osaka-u.ac.jp

very simplified network model: a single TCP connection,[7,8] only two TCP connections,[9] or homogeneous TCP connections.[10] So those analytic results are not directly applicable to a real network. In addition, stability of TCP Vegas has not been fully investigated. Since TCP Vegas is essentially a feedback-based congestion control, a stable operation of the control mechanism is of great importance.

In our previous works,[11,12] we have analyzed dynamics of the window-based flow control mechanism based on TCP Vegas using a control theoretic approach. In the first paper,[11] we have analyzed stability and transient performance of the window-based flow control mechanism. However, the analytic model is limited to a homogeneous network, where all TCP connections have identical propagation delays. In the second paper,[12] we have extended our analysis to a more generic network model, where each TCP connection is allowed to have a different propagation delay. We have derived stability condition of the window-based flow control mechanism based on TCP Vegas. We have quantitatively shown how stability of the window-based flow control mechanism is affected by several network parameters. However, we have investigated only the simplest case; i.e., when the propagation delay ratio of TCP connections is 1:2. In this paper, using our analytic results, we investigate the dynamics of the window-based flow control mechanism in realistic network configurations. Namely, we investigate stability and transient performance of the window-based flow control mechanism when the propagation delay ratio of TCP connections is changed.

Organization of this paper is as follows. In Section 2, we explain the congestion avoidance mechanism of TCP Vegas, and introduce our analytic model. In Section 3, we briefly summarize a control theoretical analysis for stability and transient performance, which is an excerpt from our previous works.[12] In Section 4, through numerical examples and simulation results, the effect of control parameters of the window-based flow control mechanism on stability is investigated. We also discuss how to configure control parameters of the window-based flow control mechanism for achieving the best transient performance. In Section 5, 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, multiple TCP connections are established through a single bottleneck router. There are M groups of TCP connections, where in each group TCP connections have an identical propagation delay. The number of connections in group m is denoted by N_m. Let τ_m be the propagation delay of a TCP connection 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 as the ratio of propagation delays τ_m's. Namely,

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

By assuming that the waiting time of a packet at the router's buffer is negligible, the ratio of the RTT for a TCP connection in group m is given by Δ_m. In TCP Vegas, a source host changes its window size once per RTT.[3] Therefore, the system can be modeled as a discrete-time system where the time slot is given by τ_m/Δ_m. In other words, a TCP connection in group m changes its window sizes once every Δ_m slots.

Let $w_{m,n}(k)$ be the window size of n-th TCP connection $(1 \leq n \leq N_m)$ in group m at slot k. We define $q(k)$ as the number of packets in the router's buffer at slot k. We define L and B as the buffer capacity of the router and the processing speed of the bottleneck router, respectively. Then, the window size at slot $k + \Delta_m$ and the number of packets in the router's buffer at slot $k + 1$ are given by the following equations.[12]

$$w_{m,n}(k + \Delta_m) = \max\left\{w_{m,n}(k) + \delta_{m,n}(\gamma_{m,n} - d_{m,n}(k)), 0\right\} \quad (1)$$

$$q(k+1) = \min\left[\max\left\{\sum_{m=1}^{M}\sum_{n=1}^{N_m}\left(w_{m,n}(k) - \frac{w_{m,n}(k) B \Delta_m \tau}{\sum_{m=1}^{M}\sum_{n=1}^{N_m} w_{m,n}(k)}\right), 0\right\}, L\right] \quad (2)$$

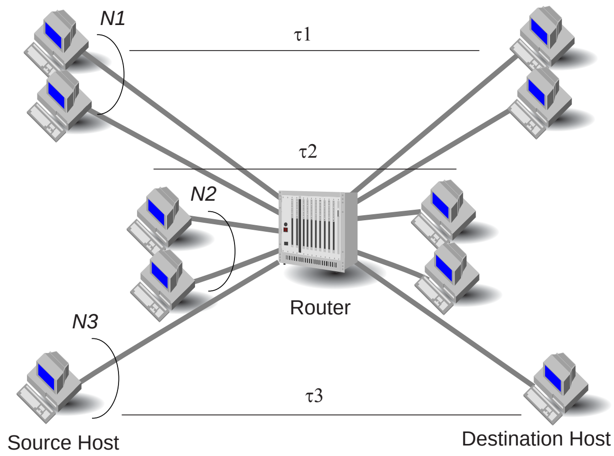


Figure 1: Analytic model for $M = 3$.

where

$$d_{m,n}(k) = \left(\frac{w_{m,n}(k)}{\tau_m} - \frac{w_{m,n}(k)}{r_m(k)} \right) \times \tau_m \tag{3}$$

$$r_m(k) = \tau_m + \frac{q(k)}{B} \tag{4}$$

In the above equation, both γ_m and $\delta_{m,n}$ are control parameters at the source host. The parameter γ_m controls the number of excess packets in the network sent from a source hosts in group m. The parameter $\delta_{m,n}$ adjusts the amount of the window size change per RTT. Note that Eq. (4) approximates RTT measured by a source host in group m at slot k. We note that Eq. (2) is changed from that of our previous analysis[12] to reduce the approximation error, in particular, when Δ_m takes a large value.

3. STABILITY AND TRANSIENT PERFORMANCE ANALYSIS

In this section, we briefly present stability and transient behavior analyses of the window-based flow control mechanism based on TCP Vegas. Refer to the paper[12] for the detail of these analyses.

In what follows, we assume that initial values of window sizes of all source hosts are identical, and also assume that control parameters of TCP connections in the same group are identical. For brevity, the control parameter of a source host 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. (1), the number of packets in the router's buffer at slot $k+1$ is given by

$$q(k+1) = \min \left[\max \left\{ \sum_{m=1}^{M} N_m \left(w_m(k) - \frac{w_m(k)\, B\, \Delta_m\, \tau}{\sum_{m=1}^{M} N_m w_m(k)} \right), 0 \right\}, L \right] \tag{5}$$

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), (2), and (3), w_m^*, q^* and d_m^* can be easily obtained. Let $\mathbf{x}(k)$ be the difference of the system state from its equilibrium value at slot k, i.e.,

$$\mathbf{x}(k) \equiv \begin{bmatrix} w_1(k) - w_1^* \\ w_2(k) - w_2^* \\ \vdots \\ w_{M-1}(k) - w_{M-1}^* \\ w_M(k) - w_M^* \\ q(k) - q^* \end{bmatrix}$$

Since $w_m(k)$ is a non-linear equation, we linearize it around the equilibrium 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

$$\mathbf{x}(k+\Delta_L) = \mathbf{A}\,\mathbf{x}(k) \tag{6}$$

where $\mathbf{A}$ is a state transition matrix. Stability and transient behavior of the system around its equilibrium are determined by eigenvalues of the state transition matrix $\mathbf{A}$. More specifically, the equilibrium point is locally asymptotically stable if and only if all roots $s_i (1 \le i \le M+1)$ of the characteristic equation $D(s) = |sI - A| = 0$ lie in the unit circle.[13] It can be easily shown by using the Jury's criterion whether the state transition matrix $\mathbf{A}$ satisfies this condition or not.[13] In addition, it is known that the smaller absolute values of eigenvalues are, the better the transient performance becomes.[14] It is also known that transient performance of the system is mostly determined by the maximum value of λ_i.

$$|\lambda| \equiv \max_i |\lambda_i| \tag{7}$$

In the following sections, $|s|$ is referred to as a *transient performance index*.

4. NUMERICAL EXAMPLES

In this section, we show several numerical examples for $M = 2$, i.e., the case of two TCP connections with different propagation delays. The main purpose is to quantitatively show how the difference in propagation delays affects stability and transient performance of the network.

4.1. Discussion of Stability

The window-based flow control mechanism discussed in this paper may result in an unstable operation unless its control parameters are chosen appropriately.[12] If the network is unstable, the window size of a source host and the number of packets in the router oscillate and never converge to equilibrium values. Hence, it is important to carefully configure control parameters of the window-based flow control mechanism. In our previous work,[12] we have shown a parameter configuration guideline for the simplest network configuration, i.e., $\Delta_1 = 1$ and $\Delta_2 = 2$. The result suggests that, for a stable operation of the network, the control parameter δ_m, which adjusts the amount of a window size change per RTT (i.e., the feedback gain), should be proportional to the propagation delay of each TCP connection. The main objective of this section is to investigate more generic cases.

After investigating stability region (δ_1, δ_2) for a number of parameter sets (i.e., the processing speed of the bottleneck router B, the control parameter γ_m, and the propagation delay τ_m), we have found that the stability region (δ_1, δ_2) is unchanged if the following value is unchanged.

$$F_m \equiv \frac{N_m\,\gamma_m}{B\,\tau} \tag{8}$$

In what follows, we therefore discuss based on this value F_m. Intuitively, F_m has the following meaning. As Eq. (1) indicates, the source host adjusts its window size to send the number γ_m of extra packets into the network per its

RTT. Provided that the number of packets conveyed on transmission links is sufficiently smaller than the number of packets in the router's buffer, the number of packets in the router's buffer can be approximated by the following equation.[12]

$$q^* \simeq \sum_{m=1}^{M} \sum_{n=1}^{N_m} \gamma_{m,n} \quad (9)$$

$$= \sum_{m=1}^{M} N_m \gamma_{m,n} \quad (10)$$

Since the packet waiting time in the router's buffer is given by q^*/B, F_m can be though of as the packet waiting time of a TCP connection, divided by its propagation delay. Namely, F_m can be thought of as a value representing the amount of the packet waiting time in the router's buffer compared to the round-trip time. For instance, if F_m is small, the packet waiting time in the router's buffer is relatively small than the propagation delay of the TCP connection. On the contrary, if F_m is large, the packet waiting time in the router's buffer is larger than the propagation delay of the TCP connection.

We first show stability regions (δ_1,δ_2) for the propagation delay ratio 1:4 in Fig. 2, and for the propagation delay ratio 2:3 in Fig. 3. In these figures, the network is stable if the point (δ_1,δ_2) lies within the boundary line. In these figures, F_m is changed from 0.015 to 150 while satisfying the relation $F_1 = F_2$. These figures suggest, when F_m is small, the maximum value of δ_2 is larger than that of δ_1 (note $\Delta_1 < \Delta_2$ in both figures). When F_m is small, the measured RTT at source host is almost determined by the propagation delay. Hence, as well as the case of the propagation delay ratio 1:2,[12] the maximum value of δ_m becomes almost proportional to the propagation delay ratio Δ_m. On the other hand, when F_m is large, the maximum value of δ_m for achieving stability becomes independent of the difference in propagation delays of TCP connections. Figures 2 and 3 indicate, for a large F_m, the stability region is roughly $0 < \delta_1, \delta_2 < 2$. From these observations, when F_m is large (i.e., the packet waiting time is large), the network can be stabilized with, for example, (δ_1, δ_2) = (1, 1). However, when F_m is small (i.e., the packet waiting time is small), δ_m must be chosen carefully according to the difference in propagation delays of TCP connections.

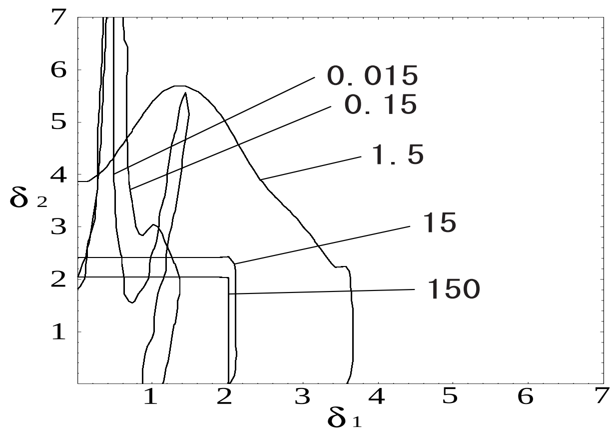


Figure 2: Stability region for propagation delay ratio 1:4 ($F_1 = F_2$ = 0.015—150, $\Delta_1 = 1$, $\Delta_2 = 4$)

4.2. Discussion of Transient Behavior

For an efficient operation of the window-based flow control mechanism, control parameters should be configured by taking account of transient performance as well as stability. In what follows, we investigate how control parameters

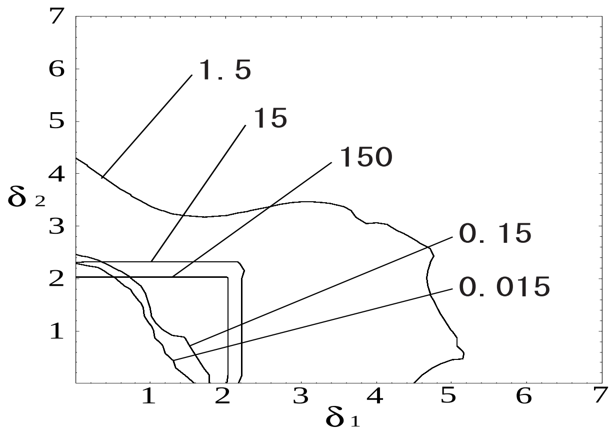


Figure 3: Stability region for propagation delay ratio 2:3 ($F_1 = F_2$ = 0.015—150, $\Delta_1 = 2$, $\Delta_2 = 3$)

should be chosen to optimize the transient performance while satisfying stability of the network. Namely, we derive the optimal values of δ_m and γ_m to minimize the rise-time, i.e., the time taken for the window size of a source host and the number of packets in the router's buffer to converge to their equilibrium values. As discussed in Section 3, stability of the network is mostly determined by the transient performance index $|s|$. Note that the actual transient performance is determined not only by the value of $|s|$ but also by the length of $\Delta_L \tau$ since the window size and the number of packets in the router's buffer change per $\tau \Delta_L$ in Eq. (6). Thus, even when $|s|$ is the same, the ramp-up time is different for a different value of $\Delta_L \tau$.

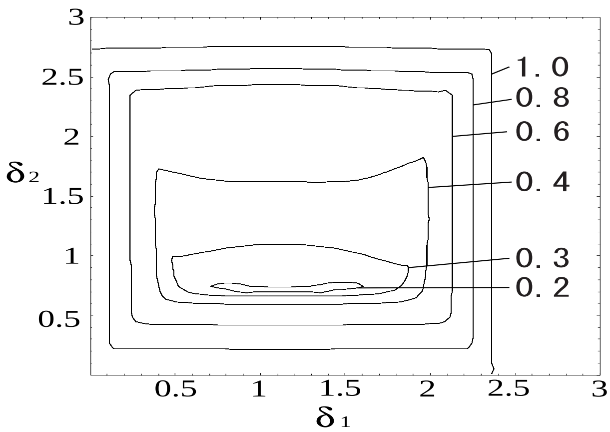


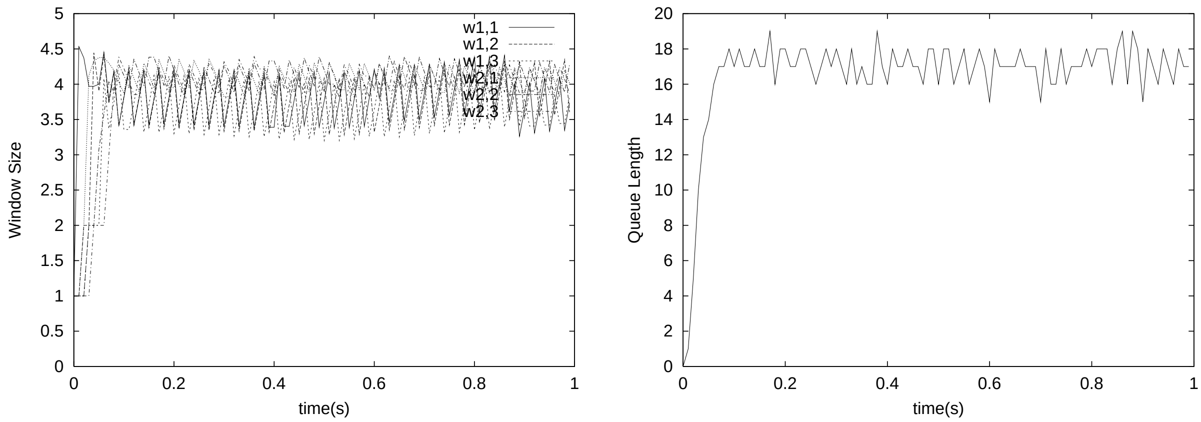
Figure 4: Contour of transient performance index $|s|$ in the δ_1–δ_2 plane ($F_1 = F_2 = 4.5$, $\Delta_1 = 1$, $\Delta_2 = 2$)

Figure 4 shows a contour of the transient performance index $|s|$ on the δ_1–δ_2 plane. In this figure, the propagation delay ratio is 1:2, i.e., $\Delta_1 = 1$ and $\Delta_2 = 2$. The following parameters are used: the processing speed of the bottleneck router $B = 2$ [packet/ms], the number of TCP connections $N_1 = N_2 = 3$, the propagation delay $\tau = 1$ [ms], the control parameter $\gamma_1 = \gamma_2 = 3$ [packet]. The value of F_m is set to $F_1 = F_2 = 4.5$. This figure indicates that, when the point (δ_1, δ_2) is outside the contour $|s| = 1.0$, the network becomes unstable. Also indicated is that the transient performance is good when the point (δ_1, δ_2) is inside the contour $|s| = 0.2$.

To clearly view how the transient performance changes according to the value of the transient performance index $|s|$, we have run simulation experiments using ns2 simulator.[15] Being different from TCP Vegas, the window-based flow control mechanism discussed in this paper has control parameters δ_m, which specifies the amount of window size change per RTT. We have therefore modified a few lines of the ns2 simulator. Table 1 summarizes settings of (δ_1, δ_2), the value of the transient performance index $|s|$, and the corresponding figure in our simulation experiments. In Figs 5 and 6, windows size of all source hosts and the number of packets in the router's buffer are plotted. The simulation model is equivalent to Fig. 1. In Fig. 5, after starting the simulation, window sizes of source hosts and the number of packets in the router's buffer quickly converge to their equilibrium values within 0.1 [s]. One would notice that window sizes of source hosts and the number of packets in the router's buffer slightly oscillate even in steady state. This is probably because of a timer granularity of TCP.[11] On the other hand, in Fig. 6, window sizes of source hosts and the number of packets in the router's buffer oscillate with a large amplitude, resulting in an unstable operation of the network.

Table 1. Simulation parameters

	(δ_1, δ_2)	transient performance index $|s|$	corresponding figure
optimal case	(1.3, 0.7)	0.18	Fig. 5
unstable case	(3.0, 3.0)	2.56	Fig. 6

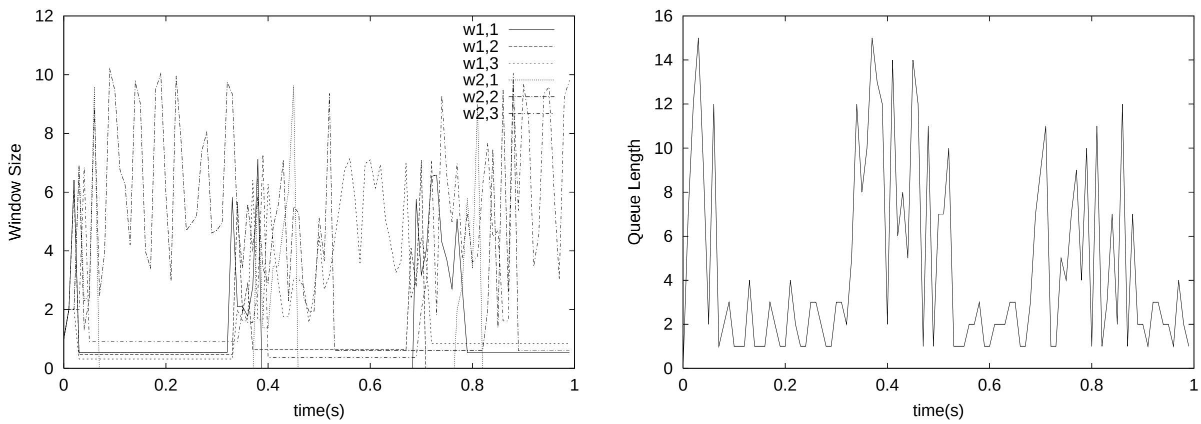


(a) Window size

(b) Number of packets in router's buffer

Figure 5: Simulation result (optimal case) ((δ_1, δ_2)=(1.3, 0.7), $F_1 = F_2 = 4.5$, $\Delta_1 = 1$, $\Delta_2 = 2$)

We next show the case of a small F_m. Figure 7 uses the same parameters with Fig. 4, but the processing speed of the router B and the propagation delay τ are changed to $B = 200$ [packet/ms] and $\tau_m = 10$ [ms], respectively. In this figure, F_m takes $F_1 = F_2 = 0.0045$ in this case. The figure shows that the transient performance index $|s|$ is quite large such as 0.992–1.0; that is, regardless of a choice of (δ_1, δ_2), the transient performance index $|s|$ is always larger than 0.99, indicating bad transient performance. To see this clearly, we choose (δ_1,δ_2)=(0.6,1.7), which is the best choice in Fig. 7, and show simulation results in Fig. 8. This figure indicates that the network takes very long time (about 10 [s] in this case) to converge to its equilibrium, even though (δ_1,δ_2) is chosen to minimize the transient performance index $|s|$. Thus, this result suggests that a large F_m is desirable for achieving the better



(a) Window size (b) Number of packets in router's buffer

Figure 6: Simulation result (unstable case) ((δ_1, δ_2)=(3.0, 3.0), $F_1 = F_2 = 4.5$, $\Delta_1 = 1$, $\Delta_2 = 2$)

transient performance.

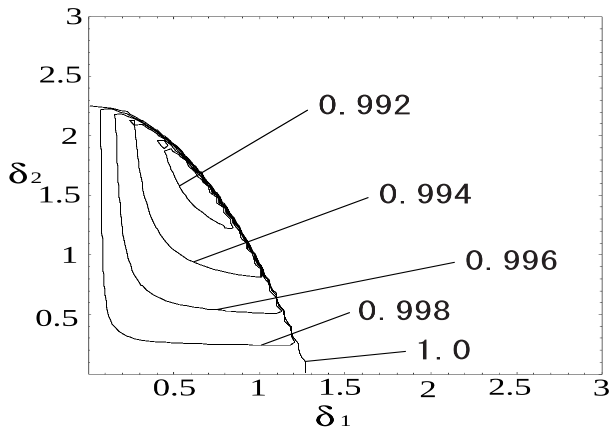
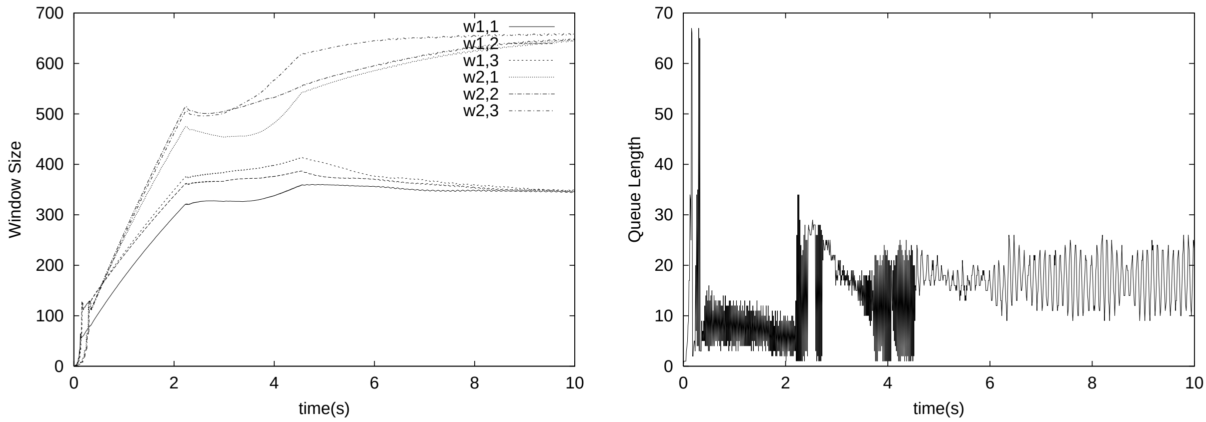


Figure 7: Contour of transient performance index $|s|$ in the δ_1–δ_2 plane ($F_1 = F_2 = 0.0045$, $\Delta_1 = 1$, $\Delta_2 = 2$)

All system parameters — the number of TCP connections N_m, the processing speed of the router B, and the propagation delay τ — are uncontrollable parameters. On the contrary, the control parameter γ_m can be chosen freely at the source host. From the above observations, we conclude that a large γ_m is desired for achieving a better transient performance. As suggested by Eq. (10), the number of packets in the router's buffer, which directly affects the packet waiting time in the router's buffer, is proportional to γ_m. Therefore, there is a trade-off between the packet waiting time in the router's buffer and the transient performance. As we have discussed earlier, if the control parameter γ_m is set to a sufficiently large value, the network can be stabilized with $0 < \delta_m < 2$, which is independent of the difference in propagation delays of TCP connections.



(a) Window size

(b) Number of packets in router's buffer

Figure 8: Simulation result (optimal case) ((δ_1, δ_2)=(0.6, 1.7), $F_1 = F_2 = 0.0045$ $\Delta_1 = 1$, $\Delta_2 = 2$)

5. CONCLUSION

In this paper, we have analyzed a window-based flow control mechanism based on TCP Vegas in a heterogeneous network, where each TCP connection has a different propagation delay. First, we have modeled both TCP connections and the bottleneck router as a discrete-time system, and have derived state transition equations representing dynamics of the window-based flow control mechanism. Using a control theoretic approach, we have analyzed stability and transient performance of the window-based flow control mechanism. We have quantitatively shown the optimal parameter configuration of the window-based flow control mechanism for achieving both stability and good transient performance. We have found that stability and transient performance are heavily dependent on the ratio of propagation delays of TCP connections. We have also found that the control parameter γ_c is a key for achieving better transient performance.

Our future work is to investigate the dynamics of the window-based flow control mechanism, when a new TCP connection is established. Our ongoing research is to analyze more generic network configurations where there exist more than two bottleneck routers in the network.

REFERENCES

1. V. Jacobson and M. J. Karels, "Congestion avoidance and control," in *Proceedings of SIGCOMM '88*, pp. 314–329, Nov. 1988.
2. W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, New York: Addison-Wesley, 1994.
3. 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, Oct. 1994.
4. J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan, "Evaluation of TCP Vegas: emulation and experiment," *ACM SIGCOMM Computer Communication Review* **25**, pp. 185–195, Aug. 1995.
5. P. Danzig, Z. Liu, and L. Yan, "An evaluation of TCP Vegas by live emulation," 1995.
6. U. Hengartner, J. Bolliger, and T. Gross, "TCP Vegas revisited," in *INFOCOM (3)*, pp. 1546–1555, 2000.
7. O. A. Hellal and E. Altman, "Analysis of TCP Vegas and TCP Reno," in *Proceedings of IEEE ICC '97*, June 1997.
8. A. Kumar, "Comperative performance analysis of versions of TCP in a local network with a lossy link," *IEEE/ACM Transactions on Networking* **6**, pp. 485–498, Aug. 1998.

9. 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*, Mar. 1999.
10. T. Bonald, “Comparison of TCP Reno and TCP Vegas via fluid approximation,” Nov. 1998.
11. 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, Aug. 1999.
12. K. Takagaki, H. Ohsaki, and M. Murata, “Stability analysis of a window-based flow control mechanism for TCP connections with different propagation delays,” in *Proceedings of INET 2000: The Internet Global Summit*, July 2000.
13. R. Isermann, *Digital control systems, Volume 1: fundamentails, deterministic control*, Springer-Verlag Berlin Heidelberg, 1989.
14. G. F. Franklin and J. D. Powell, *Digital Control of Dyanamic Systems*, Addision-Wesley Publishing Company, 1980.
15. “The network simulator.” available at `http://www.isi.edu/nsnam/ns/`.