Stability Analysis of Window-Based Flow Control Mechanism in TCP/IP Networks

Hiroyuki Ohsaki, Masayuki Murata, Toshimitsu Ushio, and Hideo Miyahara
Graduate School of Engineering Science,
Osaka University
Toyonaka, Osaka, 560-8531 Japan

Abstract

Recently, a new version of the TCP mechanism, called TCP Vegas, is proposed, and has potential to achieve better performance than current mechanisms such as TCP Tahoe and Reno. This paper considers a window-based flow control mechanism based on a slightly modified congestion avoidance mechanism of TCP Vegas, and introduces its mathematical model described by a nonlinear difference equation. We calculate its fixed point, and a stability condition of the fixed point is derived.

1 Introduction

Explosive growth of computer networks has brought severe congestion problems[1], and reactive flow control and congestion control is needed to regulate transmission of information from source hosts to destination hosts. Many mechanisms such as window-based control[2] and rate-based control[3] have been proposed for avoiding congestion, and a feedback configuration is essential to provide efficient data transfer services[4, 5, 6, 7, 8]. It is expected that control theory gives a new insight in congestion control, and its applications have been studied. Keshav[4] proposed a stable control scheme using fuzzy logic based estimators. Fendrick[5] formulated delayed feedback scheme, and analyzed its stability. In ATM networks, many control-theoretic approaches have been found in [9, 10, 11, 12, 13]. Moreover, a game theoretic approach[14, 15] and a hybrid approach combining a traditional control technique and fuzzy neural networks[16] are also discussed. However, almost all control-theoretic approaches have adopted rate-based congestion control mechanisms.

On the other hand, a window-based flow control mechanism called TCP(Transmission Control Protocol)[17] has been widely used in current packet-switched networks such as TCP/IP networks. There are several versions of TCP such as Reno, Tahoe, and Vegas, and performance evaluations of these versions have been reported[18, 19, 20, 21, 22]. TCP Reno implemented in BSD UNIX uses packet losses in the networks as feedback information, and controls a window size[17]. Recently, TCP Vegas has been proposed in order to improve performance such as throughput[18, 19, 20], and 37-71% improvement of total throughput and about 1/5-1/2 reduction of the number of retransmitted packets have been achieved. A major reason for such good performance is an introduction of a new congestion avoidance mechanism.

In this paper, we discuss stability of a window-based flow control based on TCP Vegas. Section 2 introduces a mathematical model of a network with a window-based flow control mechanism. Section 3 gives a stability condition of a fixed point in the proposed model.

2 Mathematical Model

We consider a network model shown in Fig. 1 where N source hosts are connected to the corresponding destination hosts through one router. We introduce a mathematical model of the network. Let $w_n(k)$ be a current window size of the host $n (1 \leq n \leq N)$ at slot k, and $q(k)$ the number of packets stored in router's buffer. Moreover, let L and B be the buffer's capacity and the bandwidth of the router, respectively. We assume that each source host n has always send $w_n(k)$ packets at

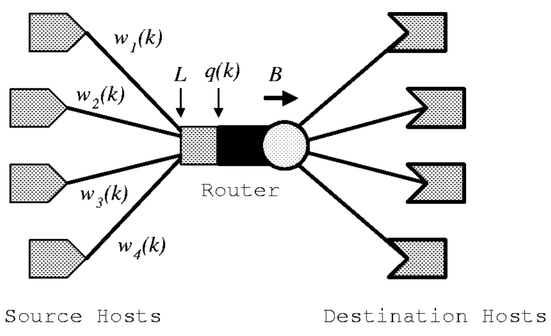

Figure 1: Network model

each slot, and that the round trip times of all connections are equal. Then, the dynamics of the buffer is given by the following nonlinear difference equation:

$$q(k+1) = \min\left(\max(\sum_{n=1}^{N} w_n(k) - Br(k),\ 0), L\right) \quad (1)$$

where $r(k)$ is the round trip time at slot k.

We review a congestion avoidance mechanism of TCP Vegas briefly. Refer to [19] for more detail. We assume that each source host knows a round trip time(RTT) τ in the case that there is no congestion in the network. In practice, τ is set to be the minimum of all measured RTT's. Note that τ corresponds to the round-trip transmission delay when the buffer is empty, that is, τ is equal to the sum of all propagation delays and processing delays at the router. Then, the throughput is expected to $w_n(k)/\tau$ if the network is not congested. On the other hand, in practice, we can obtain the RTT $r(k)$ at slot k by measuring time delay from sending a packet to receiving its ACK packet, and it is given by the following equation in our mathematical model.

$$r(k) = \tau + \frac{q(k)}{B} \quad (2)$$

The current throughput is $w_n(k)/r(k)$, and the error $d_n(k)$ between the expected and the current throughput is calculated as follows:

$$d_n(k) = \frac{w_n(k)}{\tau} - \frac{w_n(k)}{r(k)} \quad (3)$$

Vegas controls the next window size $w_n(k+1)$ based on the error $d_n(k)$ and given control parameters α and β as follows:

$$w_n(k+1) = \begin{cases} w_n(k)+1 & \text{if } d_n(k) < \alpha \\ w_n(k)-1 & \text{else if } d_n(k) > \beta \\ w_n(k) & \text{otherwise} \end{cases} \quad (4)$$

Intuitively, these parameters α and β correspond to having too little and too much extra data in the network, respectively.

From Eq. (4), Vegas does not change the window size if $d_n(k)$ lies in the interval $[\alpha,\ \beta]$, which sometimes causes unfairness among the connections[22]. So, in our analysis, in order to improve the fairness, we introduce a new control parameter γ, and set $\gamma = \alpha = \beta$, and Eq. (4) is modified as follows:

$$w_n(k) = \max(w_n(k) + \delta(\gamma - d_n(k)),\ 0) \quad (5)$$

where δ is a control parameter in order to determine the change rate of the window size per each RTT. So it is regarded as a feedback gain.

3 Stability Analysis

For simplicity, we assume that all the initial window sizes of all hosts are equal, and all hosts change their window sizes according to Eq. (5). Then the queue length $q(k+1)$ of the buffer in the router at slot $k+1$ is given by

$$q(k+1) = \min(\max(Nw(k) - Br(k),\ 0),\ L) \quad (6)$$

where

$$w(k) \equiv w_n(k) \qquad 1 \le n \le N$$

Moreover, by Eq. (3), the error $d(k) \equiv d_n(k)(1 \le n \le N)$ is given by

$$d(k) = \frac{w(k)}{\tau} - \frac{w(k)}{r(k)} \quad (7)$$

We use the following approximation.

$$Br(k) \simeq B\tau$$

Then, using Eqs. (5), (6), and (7), a fixed point $(w^*,\ q^*,\ d^*)$ of the considered network is given as follows:

$$\left.\begin{array}{rcl} w^* &=& \tau\left(\frac{B+\gamma N}{N}\right) \\ q^* &=& \gamma N \tau \\ d^* &=& \gamma \end{array}\right\} \quad (8)$$

The linearized equation around the fixed point is given by

$$x(k+1) = Ax(k) \quad (9)$$

where

$$\begin{array}{rcl} x(k) &=& \begin{bmatrix} w(k) - w^* \\ q(k) - q^* \end{bmatrix}, \\ A &=& \begin{bmatrix} 1 - \frac{\delta}{\tau} + \frac{B\delta}{(B+\gamma N)\tau} & -\frac{B\delta}{N(B+\gamma N)\tau} \\ N & 0 \end{bmatrix} \end{array}$$

Using a stability test such as Jury's test[23], it is easily shown that the fixed point is locally exponentially stable if and only if the following inequalities hold.

$$\delta > 0 \quad (10)$$

$$\frac{\delta(B-\gamma N)}{(B+\gamma N)\tau} + 2 > 0 \quad (11)$$

$$\frac{B\delta}{(B+\gamma N)\tau} < 0 \quad (12)$$

Thus, we have the following conclusion.

- If the number N of connections is greater than $3B/\gamma$, then the upper bound of the control parameter δ is given by Eq. (11). So, as N increases, the upper bound decreases monotonically, and converges to 2τ. Therefore, in this case, the network is always stable if δ is set to be 2τ.

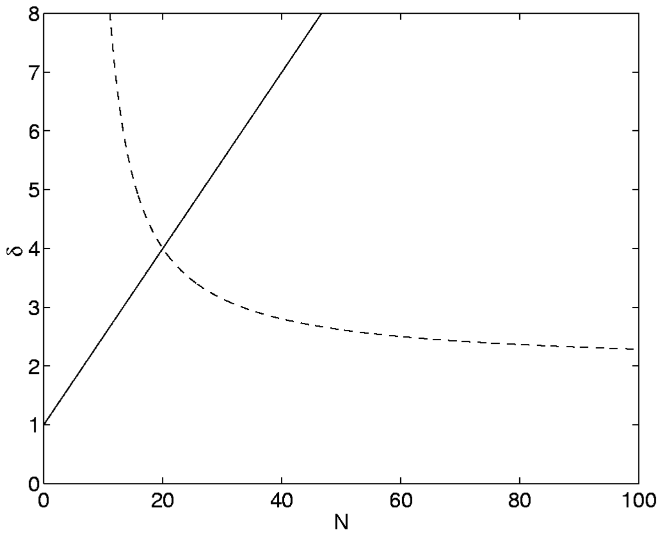

Figure 2: Stability region in the δ-N plane

- If N is less than $3B/\gamma$, then the upper bound is given by Eq. (12). So the upper bound increases linearly as N increases. Since δ is bounded by τ at $N = 0$, the network is always stable if δ is set to be τ.

Shown in Fig. 2 is a stability region where B = 20[packets/ms], $\tau = 1$[ms], and $\gamma = 3$. Note that $B = 20$ corresponds to 163.8[Mbit/s] if the packet length is equal to 1 [Kbyte]. In this figure, the dotted and solid lines correspond to Eqs. (11) and (12), respectively. When the number N of connections are less than 20, δ can be taken larger as N increases. But when it is greater than 20, δ should be smaller as N increases. It is shown that the network is always stable for any $N > 0$ if δ is less than one. Moreover, since the dotted line converges to $\delta = 2$, for sufficiently large number of connections, we take δ less than 2 for the network to be stable. From Eq. (5), Vegas' control mechanism is regarded as static feedback with feedback gain δ from the control theoretical point of view. So, in general, if δ is set to be larger, that is, we use high gain feedback, then transient behaviors of the network will be better while stability will be worse. Figure 2 shows such a tendency.

Shown in Figs. 3 and 4 are behaviors of the window size and the queue length for $N = 10$ when the network are stable and unstable, respectively. When the network is stable, both the window size and the queue length converge to the fixed point given by Eq. (8). However, when it is unstable, they oscillate with large amplitude, which implies that transmission delay of packets becomes very large. Moreover, the buffer becomes empty periodically, which implies that the throughput of the network degrades. Thus, it is concluded that δ has to be selected such that the network is stable.

4 Conclusion

This paper derived a stability condition of a window-based flow control mechanism based on a slightly modified congestion avoidance mechanism of TCP Vegas. In the case that the number of connections is small, a control parameter can be set to be larger as the number increases while the reverse feature occurs in the case that the number is large.

It is a future work to extend our analysis to more realistic networks. For example, both TCP and UDP traffic coexist in real TCP/IP networks, and interference between different types of traffic should be taken into consideration.

References

[1] V. Jacobson and M. J. Karels, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM'88*, pp. 314–329, Aug. 1988.

[2] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[3] W. Stallings, *High-Speed Networks :TCP/IP and ATM Design Principles*, Prentice-Hall, Upper Saddle River, NJ, 1998.

[4] S. Keshav, "A Control-theoretic approach to flow control," *Proc. ACM SIGCOMM'91*, pp. 3–15, Sept. 1991

[5] K. W. Fendrick, "Analysis of a rate-based control strategy with delayed feedback," *Proc. ACM SIGCOMM'92*, pp. 136–148, 1992.

[6] L. Benmohamed and S M. Meerkov, "Feedback control of congestion in packet switching networks: the case of a single congested node," *IEEE/ACM Trans. Networking*, vol. 1, no. 6, pp. 693–708, Dec. 1993.

[7] S. Fuhrmann, Y. Kogan, and R. A. Milito, "An adaptive autonomous network congestion controller," *Proc. 35th IEEE CDC*, pp. 301–306, Dec. 1996.

[8] U. Madhow, "Dynamic congestion control and error recovery over a heterogeneous internet," *Proc. 36th IEEE CDC*, pp. 2368–2374, Dec. 1997.

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

[10] C. E Rohrs and R. A. Berry, "A linear control approach to explicit rater feedback in ATM networks," *Proc. IEEE INFOCOM'97*, pp. 277–282, April 1997.

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

[12] Y. Zhao, S. Q. Li, and S. Sigarto, "A linear dynamic model for design of stable explicit-rate ABR control," *Proc. IEEE INFOCOM'97*, pp. 283–292, April 1997.

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

[14] E. Altman and T. Başar, "Multi-user rate-based flow control," *IEEE Trans. Communications*, vol. 46, no. 7, pp. 940–949, July 1998.

[15] R. T. Maheswaran and T. Başar, "Multi-user flow control as a Nash game: performance of various algorithms ," *Proc. 37th IEEE CDC*, pp. 1090–1095, Dec. 1998.

[16] C. Kwan,. R. Xu, L. Haynes, C. -H. Chou, and E. Geraniotis, "Fast flow control in high-speed communication networks," *Proc. 37th IEEE CDC*, pp. 2139–2140, Dec. 1998.

[17] W. S. Stevens, *TCP/IP Illustrated Vol. 1: The Protocol*, Addison Wesley, New York, 1994.

[18] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," *ACM SIGCOMM'94*, pp. 24–35, Aug. 1994.

[19] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," *IEEE J. Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, 1995.

[20] J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan, "Evaluation of TCP Vegas: emulation and experiment," *ACM SIGCOMM'95*, pp. 185–195, Aug. 1995.

[21] K. Fall and S. Floyd, "Simulation-based comparison of Tahoe, Reno, and SACK TCP," *ACM Computer Communication Review*, vol. 26, no. 3, pp. 5–21, July 1996.

[22] 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.

[23] C. L. Phillips and H. T. Nagle, Jr., *Digital Control System Analysis and Design*, Prentice-Hall Englewood Cliffs, NJ, 1984.

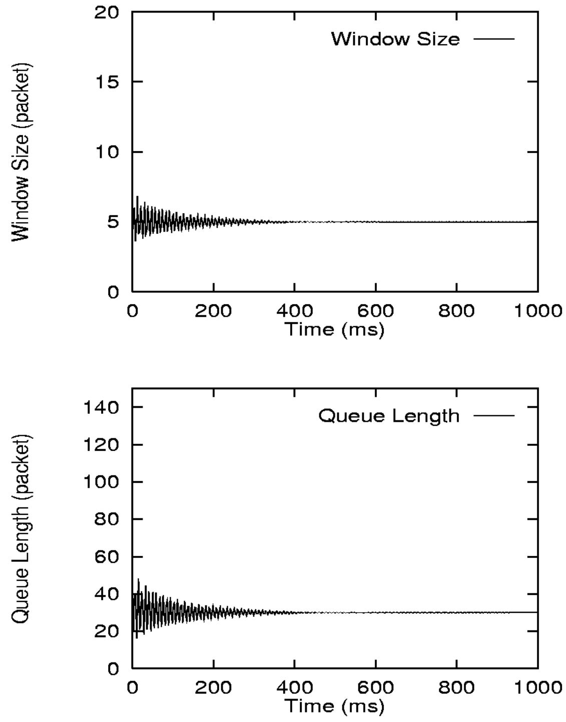

Figure 3: Stable behavior($\delta = 2.4$)

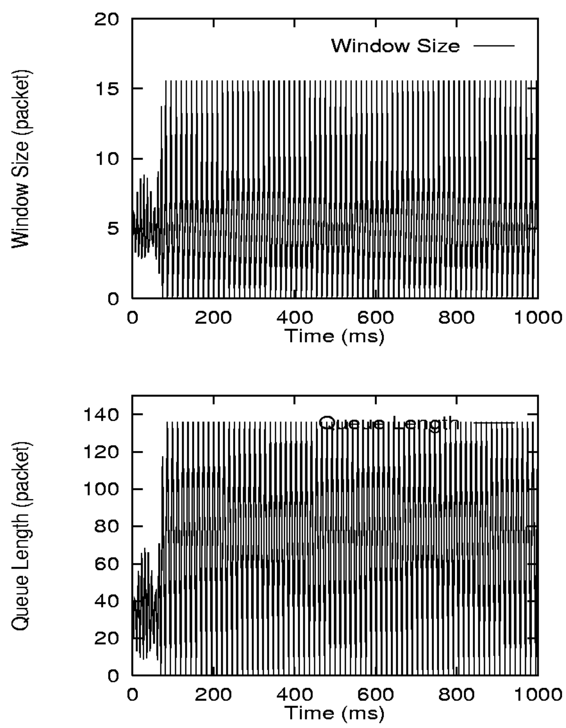

Figure 4: Unstable behavior($\delta = 2.6$)