Active Queue Management in TCP Networks Based on Minimum Variance Adaptive Control

Mohammad Shakiba Herfeh and Mohammad Haeri, *Member, IEEE*

Abstract—**Active Queue Management (AQM) is an effective method to enhance the congestion control. In this paper a control theoretic method is used to achieve a better performance in AQM in TCP networks. Using a parameter estimation scheme, a minimum variance (MV) controller is implemented in order to minimize the fluctuation of queue length which in turn results in minimum jitter (the variances of delay). We illustrate our results using *ns-2* simulations and demonstrate the practical impact of Minimum Variance Adaptive Control (MVAC) on controlling jitter and queue utilization.**

I. INTRODUCTION

Queue management has played an important role in the congestion control in the recent years. Several AQM schemes have so far been proposed which are based on either control theory or heuristic methods [1]-[4]. The two main objectives of a queue management scheme, i.e. high link utilization and low packet queuing delay, are often in conflict with each other. A low queue means low resource utilization and a high one leads to long delays.

The traditional scheme for buffer management is First-In-First-Out (FIFO) drop-tail, which drops packets only when the buffer overflows take place. Drop-tail, as the simplest and most commonly used algorithm in the current Internet routers, drops the packets from the tail of a completely full buffer of queue. TCP sources with drop-tail queue in routers reduce their sending rates only after a queue overflow is detected. This passive behavior along with the longer queuing delay often causes correlation among packet drops, resulting in the TCP synchronization problem [5] and, consequently, low link utilization.

The Random Early Detection algorithm (RED) was first proposed by Floyd and Jacobson [2]. In the RED algorithm, average queue size is determined for each packet arrival using an Exponential Weighted Moving Average (EWMA) window. The EWMA queue size is compared with a minimum and a maximum threshold, to determine the next action of the router. The main disadvantage of RED algorithm is that its performance is very sensitive to the parameters settings and a badly configured RED router will not do better than a drop-tail router. Some other heuristic AQM algorithms such as BLUE [6], DRED [7] and SRED [8] have been proposed to modify RED.

Hollot *et al.* [4] designed an AQM based on the classical control theory and the dynamic model of the TCP congestion control. They assume that the AQM mechanism brings the system to the neighborhood of an equilibrium (operating point), so one can use the approximated linear model to design the controller. The control parameters are determined solely by the network and traffic conditions, and under certain configurations sluggish responses would occur. Large deviations from the operating point would generally result in unacceptable results, although the system is stable.

In this paper, we propose Minimum Variance Adaptive Control (MVAC) algorithm, to improve the link utilization and achieve a good jitter parameter. Jitter, which is an important concept in the network literature, means the variation of delays in data transmission. The main goal of our algorithm is to make the queue track the desired queue length having the smallest variations around it.

The remainder of this paper is organized as follows. The MVAC and its algorithm are described in Section II. In Section III, performance of the proposed controller is evaluated through some *ns-2* simulations and compared to that of some existing AQM schemes. Finally Section IV concludes the paper.

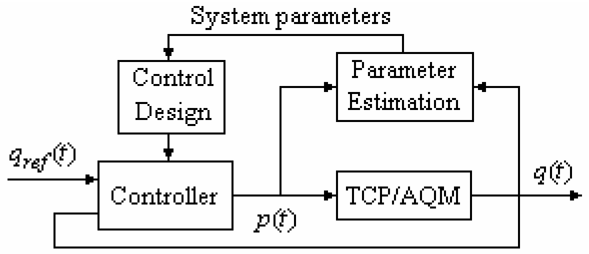

Fig. 1. The Minimum Variance Adaptive Control Architecture

II. MINIMUM VARIANCE ADAPTIVE CONTROL

A. Parameter Estimation

The PI controller designed by Hollot *et al.* [4] is based on a fixed mathematical model of TCP network, which assumes that some of the network parameters such as number of TCP sessions, Round Trip Time (RTT), and bandwidth are fixed. But in the real network these parameters varies time to time.

Manuscript received November 14, 2005. This work was supported in part by the Iranian Telecommunication Research Center (ITRC) under Grant T500/10177.

Mohammad Shakiba Herfeh is with the Advanced Control System Lab, Electrical Engineering Department, Sharif University of Technology, Azadi Ave., Tehran, Iran; (E-mail: shakiba@mehr.sharif.edu).

Dr. M. Haeri is with the Electrical Engineering Department, Sharif University of Technology, and the head of the Advanced Control System Lab. Tel: +98 (21) 6616 5964, Fax: +98 (21) 6602 3261. (E-mail: haeri@sina.sharif.edu).

In order to overcome this varying parameter problem we propose on line parameter estimation in AQM.

The MVAC architecture is shown in Fig. 1. Based on the input and output measurements, the model parameters are estimated online using a recursive parameter estimator. The estimated parameters are then used to calculate the control parameters according to the given design method. The controller determines the probability of packet drop/mark.

We chose ARMAX model structure for the TCP/AQM dynamics with the order of 1:

$$A(q)q(t) = B(q)p(t) + C(q)e(t) \tag{1}$$

In which $q(t)$ is the queue length and $p(t)$ is the probability of packet drop. We chose $A(q)$ and $B(q)$ as follows:

$$A(q) = q + a, \quad B(q) = b, \quad C(q) = q + c \tag{2}$$

The TCP/AQM model in (1) can be expressed as:

$$q(t) = \varphi^T \theta + \varepsilon(t) \tag{3}$$

In which:

$$\varphi(t) = \begin{bmatrix} -q(t-1) & p(t-1) & \varepsilon(t-1) \end{bmatrix}^T, \quad \theta = \begin{bmatrix} a & b & c \end{bmatrix}^T \tag{4}$$

And the parameter update rule is [9]:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + \frac{\gamma\varphi(t)}{\alpha + \varphi^T(t)\varphi(t)}(q(t) - \varphi^T(t)\hat{\theta}(t-1)) \\ \varepsilon(t) &= q(t) - \varphi^T(t-1)\theta(t-1) \end{aligned} \tag{5}$$

Where $\alpha \geq 0$ and $0 < \gamma < 2$

B. *Minimum Variance Controller*

One of the most important parameters of the network is the jitter that is preferred to be small. Jitter is the variances of delay in the packet sending. We try to minimize the jitter by minimizing the variances of queue length which is in direct relation with the delay.

The cost function is [9]:

$$J = E\{\tilde{q}(t)^2\} \tag{6}$$

Where E denotes the statistical expectation and y_m is the desired output.

Consider the dynamical system in (1). Let the polynomial F of degree $d_0 - 1$ be the quotient and the polynomial G be the remainder when $q^{d_0-1}C$ is divided by A. d_0 is the relative degree of the system i.e. $d_0 = \deg A - \deg B$.

$$\frac{q^{d_0-1}C(q)}{A(q)} = F(q) + \frac{G(q)}{A(q)} \tag{7}$$

This relation can be written in the Diophantine equation form,

$$q^{d_0-1}C(q) = A(q)F(q) + G(q) \tag{8}$$

The Minimum Variance control law is:

$$B(q)F(q)p(t) = -G(q)q(t) \tag{9}$$

Usage of this controller give rises to the minimum variance of output, around zero. In order to track the desired queue length q_{ref} we use an auxiliary variable as follows:

$$\tilde{q}(t) = q(t) - q_{ref} \tag{10}$$

Therefore, the cost function will be:

$$J = E\{\tilde{q}(t)^2\} = E\{(q(t) - q_{ref})^2\} \tag{11}$$

The TCP/AQM model of (1) and (2) can be rewritten as:

$$\begin{aligned} q(t+1) - q_{ref} + aq(t) - aq_{ref} &= \\ bp(t) - (1+a)q_{ref} &+ e(t+1) + ce(t) \end{aligned} \tag{12}$$

or

$$\tilde{q}(t+1) + a\tilde{q}(t) = b(p(t) - \frac{1+a}{b}q_{ref}) + e(t+1) + ce(t) \tag{13}$$

Now $\tilde{p}$ is defined as:

$$\tilde{p}(t) = p(t) - \frac{1+a}{b}q_{ref} \tag{14}$$

Using equation (8) and (9), the drop/mark probability will be:

$$p(t) = -\frac{c-a}{b}(q(t) - q_{ref}) + \frac{1+a}{b}q_{ref} \tag{15}$$

Where the a, b, c parameters are the outputs of the estimator.

III. SIMULATIONS AND RESULTS

In this section, we investigate the performance of MVAC performance through *ns-2* [10] simulations. We also compare its performance with existing AQM schemes, in particular, PI [4] and RED [3].

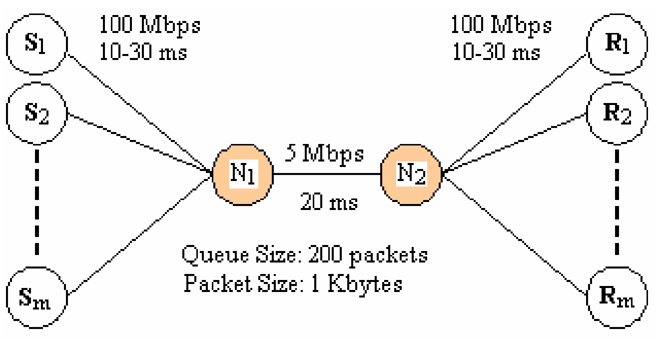

Fig. 2. The simulation topology

We implement a simple bottleneck network configuration consisting of two routers and a number of TCP senders and receivers as shown in Fig. 2. The bottleneck link has a bandwidth capacity of 5 Mbps and a buffer capacity of 200 packets. The propagation delay of the bottleneck link is set to be 20 ms and the packet length is set to 1 Kbytes. We use TCP-Reno as the default transport protocol. The propagation delays of senders are set to 10-30 ms . The reference input of the system, q_{ref} is the half of buffer size, i.e. 100 packets.

Experiment 1:

In the first experiment we use the MVAC with 200 TCP senders and receivers. We set $\gamma = 1$, and $\alpha = 0.1$, and the initial values a, b, c to 0.1, 0.2, 0.1. 200 FTP traffic sources on TCP connections will begin at time 0 and end at time 150 s .

The queue length is shown in Fig. 3 and the parameter estimation results are given in Fig.4.

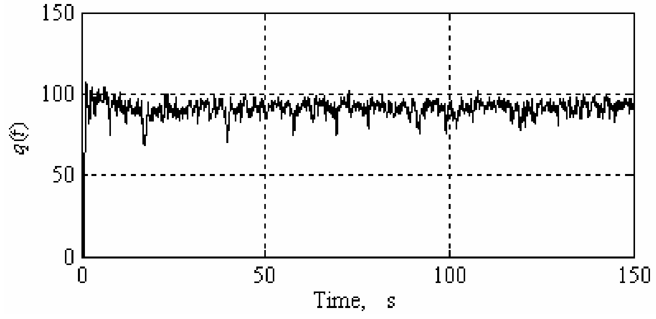

Fig.3. The queue length in the MVAC

As the network is time varying system, the parameters don’t converge to specific values and are updated to cope with the network changes.

Experiment 2:

In the second experiment we compare our proposed MVAC with some other existing controllers in AQM. The simulation topology is the same as shown in Fig. 2. 200 TCP connections are connected to the routers. 200 FTP sources start to send data at time 0. In order to change the situation, 100 FTP sources are inactivated at time 50 s and 50 of them are again activated at time 100 s.

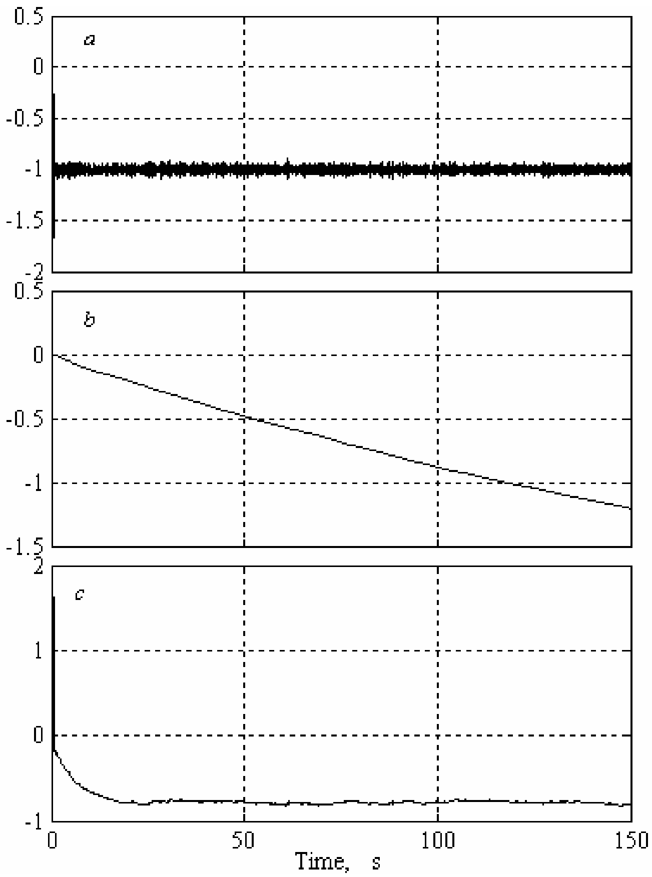

Fig.4. Parameters of the estimated ARMAX model

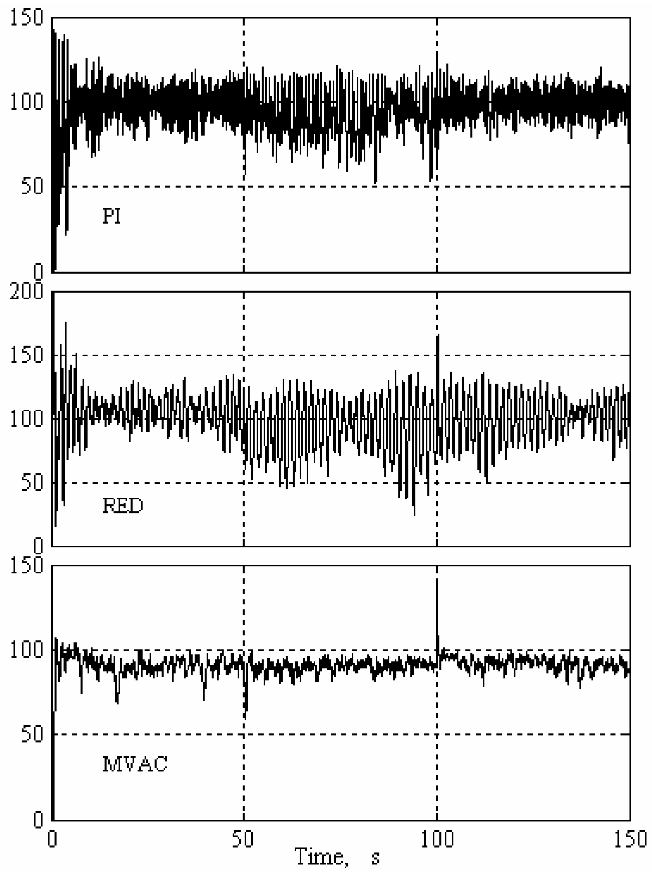

Fig.5. The queue length in different AQM controller

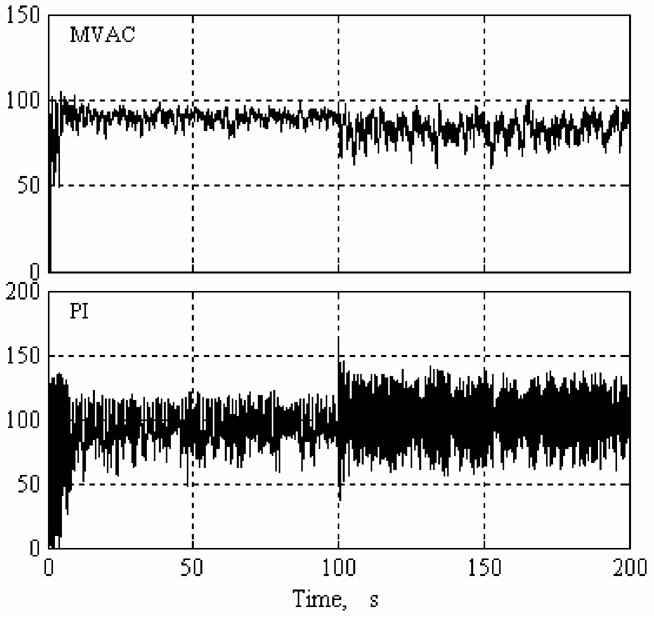

Fig.6. The queue length with the MVAC and PI controller in the presence of non-responsive flows

Fig. 5 shows the results of MVAC, PI [4], and RED [3]. As it can be noticed from the Fig. 5, RED is so sensitive to network parameter, and its performance would change according to the network parameter changes.

MVAC show better performance in minimizing the fluctuation in queue length than PI, as it is expected.

In the next experiment, the performance of PI and MVAC is compared with each other in presence of non-responsive flows.

Experiment 3:

In this experiment 10 CBR sources with the rate 0.1 Mbps begin on UDP connections at time 100 to show the system response to non-responsive traffic flows. The UDP connection is used in media communication that acknowledgment packet is not required. The results of PI controller and MVAC are shown in Fig. 6.

As it can be seen, because of online parameter estimation, the MVAC has a better performance in the presence of non-responsive flows.

IV. CONCLUSION

In this paper, we proposed a minimum variance adaptive control to apply in TCP/AQM problems. The adaptive control can achieve better performance under different traffic conditions and thus different dynamics of the system. In order to reduce jitter in the sources, minimum variance controller used to have the minimum variances of queue fluctuations.

Although MVAC algorithm is more complicated than the other AQM algorithms, the results show that it demonstrates better performance.

REFERENCES

[1] V. Misra, W. B. Gong, and D. Towsley, “Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED,” in Proc. ACM/SIGCOMM, 2000, pp. 151–160.
[2] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” IEEE/ACM Trans. Networking, vol. 1, Aug. 1993.
[3] C. Hollot, V. Misra, D. Towsley, and W. Gong, “A control theoretic analysis of RED,” In Proceedings of IEEE INFOCOM, vol. 3, pp. 1510-1519, Anchorage, Alaska, 2001.
[4] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, “Analysis and design of controllers for AQM routers supporting TCP flows,” IEEE Trans. Automat. Contr., vol. 47, pp. 945–959, June 2002.
[5] B. Braden, et al., “Recommendations on queue management and congestion avoidance in the Internet,” IETF RFC2309, Apr. 1998.
[6] W. C. Feng, K. G. Shin, D. K. Kandlur, and D. Saha, “The blue active queue management algorithm,” IEEE/ACM Trans. Networking, vol. 10, pp. 513–528, Aug. 2002.
[7] Aweya, J., Ouellette, M., and Montuno, D.Y.: ‘A control theoretic approach to active queue management’, Comput. Netw., 2001, 36, (2–3), pp. 203–35
[8] Ott, T.J., Lakshman, T.V., and Wong, L.H.: ‘SRED: stabilized RED’. Proc. IEEE/INFOCOM, 1999, pp. 1346–1355
[9] K. Astrom and B. Wittenmark, Adaptive control, 2^{ed}, Addison Wesley, N.Y., 1995.
[10] UCN/LBL/VINT, Network simulator *ns-2*, http://www.mash.cs.berkeley.edu/ns.