Performance Improvement on the Robustness of TCP/Active Queue Management with ECN

V. Santhi and A.M. Natarajan

Abstract—**Active Queue Management (AQM) can potentially reduce packet loss rate in the Internet. This is used by routers for control congestion, where packets are dropped before queues become full. It has been shown that the transmission control protocol (TCP) connections through the congested routers can be modeled as a feedback dynamic system. In this paper, a new framework of AQM, namely Variable Structure (VS) based control scheme to handle delays in active queue management algorithm supporting explicit congestion notification (ECN), is proposed. The objective of the new algorithm is to improve performance of congested routers by keeping link utilization high, stable queue size and packet drop rate low. By examining the robustness and system response for the nonlinear TCP/AQM model, we present that the proposed scheme has good performance and robustness with respect to the uncertainties of the round-trip time (RTT) and the number of active TCP connections. The planned AQM is implemented with help of ns2 simulator. The simulation shows that the proposed design outperforms the peer AQM schemes in terms of packet loss, link utilization and buffer fluctuation.**

Index Terms—**Active queue management, Congestion control, Explicit Congestion Notification (ECN), robustness, stable queue size, variable structure control.**

I. Introduction

The explosive growth of the Internet depends on the design of the best-effort service core network. The Internet is a packet switching network. Its intermediate nodes, e.g., routers, forward packets with their best efforts, but with no guarantee. Packets are forwarded on the first in first out (FIFO) strategy, and discarded when buffer overflows.

Congestion in the Internet can cause high packet loss rates, increased delays, and even break the whole system. Without congestion control, when the offered load is larger than the network capacity, the network power (ratio of throughput to delay) will decrease sharply and the network will be driven to congestion collapse.[5][6] The main targets of TCP congestion control are to explore and fully utilize the available bandwidth for a connection and to avoid severe congestions in the network.

Routers in the network are equipped with buffers to handle temporal traffic surge. The buffers may be manipulated with sophisticated mechanisms to achieve certain QoS requirements. Those mechanisms are referred to as queue management. Queue management may be classified into two types: Passive Queue Management (PQM) and Active Queue Management (AQM). PQM is activated only after the buffer is full or some threshold is exceeded. But it has two important drawbacks such as lock-out problems and always maintains the queue becomes full. To avoid this situation a new technique called Active Queue Management (AQM) implemented[1]. Different from PQM, AQM is activated before the buffer is full. By dropping packets early, it may avoid buffer from fullness. The global synchronization and lock-out problems may also be resolved with mechanisms that drop packets from different flows. In addition, by keeping low buffer occupancy, it can absorb more traffic surges.

Currently most used queue management scheme, DropTail has been observed to have some drawbacks, namely, link underutilization, queue fluctuation and global synchronization of competing TCP connections, due to its heuristic and empirical nature. Meanwhile, random early detection (RED) is an AQM mechanism proposed in [7] to reduce link congestion and global synchronization by earlier congestion notification. The main drawback of the performance of a RED router is very sensitive to link's traffic load and its parameter setting, and it is hard to reduce the queue fluctuation by only adjusting RED's parameters.

Notably, the TCP connections through the congested routers can be modeled as a feedback dynamic system, where control theory-based approaches can be used to analyze the network behavior, tune AQM's parameter settings, and design new AQM schemes. We refer to [13], [14], and [15] for the details of the nonlinear dynamic models for the TCP's additive increase and multiplicative decrease (AIMD) behavior and the AQM's congestion feedback. It is said that the control system-based analysis offers new insight into the AQM design. In [2], a control theoretic analysis was given for RED, which provided a more systematic and in-depth study on RED parameter tuning; [12] developed a PI controller as a new AQM scheme using linear system analysis; and [9] developed a VS controller without handling time delayed system as new AQM scheme. Meanwhile, explicit congestion notification (ECN) [3][4] has been proposed, where packet marking is used as congestion indication.

The aim of this paper to design a Variable Structure (VS) based control scheme to handle delays in active queue management algorithm supporting explicit congestion notification (ECN) based on combination of the queue length error and error between incoming traffic rate and link

V. Santhi is working as a Lecturer in Department of Computer Science and Engineering, PSG College of Technology, Coimbatore-641 004, TamilNadu, India. Phone:+91-98427-28113(e-mail:sannthi@yahoo.com).

Dr.A.M.Natarajan is working as CEO & Professor, Bannari Amman Institute of Technology, Sathyamangalam, Coimbatore, TamilNadu, India. (e-mail:amn@bannari.co.in)

capacity. It is evaluated using ns2 simulator. The proposed scheme has good robustness with respect to the uncertainties of the round-trip time (RTT) and the number of active TCP sessions. The simulation shows that the VS Controller to handle time delays AQM outperforms other active queue management techniques like RED, REM and PI in terms of packet loss ratio, link utilization and stable buffer space.

The rest of the paper is organized as follows. Section II describes the nonlinear TCP traffic dynamics and develops a VS-based AQM controller supporting ECN. Section III gives a description of RED, REM and PI shows why it is ineffective at managing congestion. Section IV describes evaluation of its performances based on simulations. Finally, Section V concludes with a discussion of future work.

II. Variable Structure Control to Handle Time Delays in AQM

A. Nonlinear TCP Dynamics

In [14], a nonlinear dynamic model for TCP congestion control was derived, where the network topology was assumed to be a single bottleneck with M homogeneous TCP sources that share the bottleneck link and have roughly the same RTTs, but do not necessarily transverse the same path. For TCP with ECN, the AIMD behavior in congestion avoidance phase can be modeled as follows: each positive acknowledgment increases the value of congestion window cwnd by $1/cwnd$ while each congestion indication (ECN) reduces the cwnd by half, thus, the expected change in congestion window is given by $(1-p)/cwnd - (cwnd/2)p$, where p is the marking probability. Aggregating the M TCP flows through one congested router, we have the expected change in the congestion window cwnd per update step

$$\Delta cwnd = \frac{1}{M}\left(\frac{1-p(t)}{cwnd} - \frac{cwnd}{2}\,p(t)\right) \tag{1}$$

Where p(t) is the marking probability and M is the number of the active TCP session. Denote r(t) to be the incoming traffic rate per unit time and R(t) the round trip time delay, we have $r(t) = Mcwnd/R(t)$. We assume the variation of R(t) is much slower than r(t). Since the time between update steps is about $\Delta R = R(t)/Mcwnd$, the expected change in the aggregated rate per unit time is approximately

$$\frac{\Delta r}{\Delta R} = \frac{M\Delta cwnd/R(t)}{R(t)/Mcwnd} = \frac{M}{R^2(t)} - \left(\frac{M}{R^2(t)} + \frac{r^2(t)}{2M}\right)p(t) \tag{2}$$

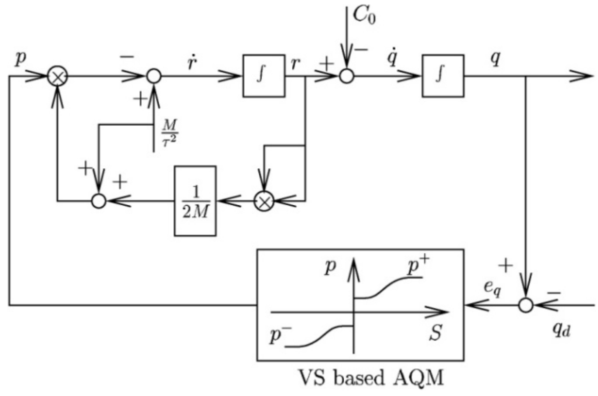

Fig. 1. Aggregated dynamics of TCP and VS-based AQM.

Motiivated by this calculation, we have

$$\dot{r}(t) = \frac{M}{R^2(t)} - \left(\frac{M}{R^2(t)} + \frac{r^2(t)}{2M}\right)p(t)$$

$$\dot{q}(t) = r(t) - Co \tag{3}$$

and

$$R(t) = \frac{q(t)}{Co} + Tp \tag{4}$$

where q(t) is the instantaneous queue length on the router, T_p is the propagation delay and C_0 is the link capacity. Note that (3) is a simplified model without considering the time delays in $r(t)$ and $p(t)$. We assume the number of active TCP connections to be uncertain, obeying

$$0 < M^- \leq M \geq M^+$$

which is more reasonable in practice.

B. VS-Based AQM with ECN to handle time delays

The nonlinear TCP dynamics and the VS-based AQM can be modeled as a feedback control system depicted in Fig. 1, where the VS controller [9] uses the queue and traffic incoming rate information to generate marking rate as congestion indication. In the dynamic TCP networks, queue size will be changed dynamically. Because of time delays in the network incoming traffic rate r(t) will be decreased and round trip time R(t) will be increased. Round Trip Time (RTT) in the network is not constant. In order to handle time delays, VS controller can be modified in the sense of Round trip time R(t) and incoming traffic rate r(t).

From (4) RTT is given as

$$R(t) = \frac{q(t)}{Co} + Tp$$

Taking changes in RTT from (4)

$$\dot{R}(t) = \frac{\dot{q}(t)}{Co}$$

(5)

and to handle time delays change in incoming traffic rate can be taken as

$$r(t) = Mcwnd/(\dot{R}(t) * Tp)$$

(6)

where $\dot{R}(t)$ is change in Round Trip Time (RTT) and Tp is propagation delay.

Let $eq = q(t) - qd$ and denote $x1 = eq$, $x2 = \dot{eq}$, where qd is the desired queue length.

We have

$$x2 = \dot{eq} = \dot{q}(t) = r(t) - Co$$

(7)

and the plant (3) can be described as

$$\dot{x}1 = x2$$

$$\dot{x}2 = \frac{M}{\dot{R}^2(t)} - \left(\frac{M}{\dot{R}^2(t)} + \frac{r^2(t)}{2M}\right)p(t)$$

(8)

Note that (8) is an affine nonlinear system in the form of

$$\dot{x} = f(x,t) + B(x,t)p(t)$$

where

$$x = \begin{bmatrix} x1 \\ x2 \end{bmatrix} \quad f(x,t) = \begin{bmatrix} x2 \\ \frac{M}{\dot{R}^2(t)} \end{bmatrix}$$

And

$$B(x,t) = \begin{bmatrix} 0 \\ -\left(\frac{M}{\dot{R}^2(t)} + \frac{r^2(t)}{2M}\right) \end{bmatrix}$$

Thus, the equivalent control method (ECM) can be used to construct the VS control law.We select the sliding mode surface as

$$S(x,t) = Kx1 + x2, \quad K > 0$$

(9)

which corresponds to a linear combination of the queue length error and the error between incoming traffic rate and link capacity.

The corresponding existence condition [16] for the sliding mode is

$$D = -\frac{GB + (GB)^T}{2} > 0$$

(10)

where $G = \partial S/\partial x$, Observe

$$b(x2) = \frac{M}{\dot{R}^2(t)} + \frac{r^2(t)}{2M} > 0$$

And

$$GB = \left[\frac{\partial S}{\partial x1} \; \frac{\partial S}{\partial x2}\right]\begin{bmatrix} 0 \\ -b(x2) \end{bmatrix} = -b(x2) < 0$$

It is straightforward that (10) is satisfied. The typical sliding mode controller [16] is given by

$$p = \bar{\propto} \; F(x,t) sign(S), \qquad \bar{\propto} > 1$$

(11)

where,

$$p_{eq} = -(GB)^{-1}Gf = \frac{Kx2 + \frac{M}{\dot{R}^2(t)}}{b(x2)}$$

(12)

And F(x,t) is any continuous function satisfying $F(x,t) > |p_{eq}(x,t)|$

Unfortunately, controller (11) is not practically feasible because the negative part of p(t) in (11) is out of the bound of $0 \le p(t) \le 1$. In what follows, we would like to improve the sliding mode controller by introducing a feedback term. Define

$$\bar{b}(x2) = \frac{\bar{M}}{\dot{R}^2(t)} + \frac{r^2(t)}{2M}$$

(13)

where $\bar{M} = (M^- + M^+)/2$ is the nominal values of M. According to the sliding mode existence condition and the robustness criteria, we construct the VS controller as following:

$$p(x,t) = \frac{\bar{M}}{\dot{R}^2(t)\bar{b}(x2)} + \left(\propto \frac{T_p}{\sqrt{2}} K\left|\frac{x2}{r(t)}\right| + \delta\right) sign(S(x,t))$$

$$= \begin{cases} p^+(x,t) & if\ S(x,t) > 0 \\ p^-(x,t) & if\ S(x,t) < 0 \end{cases}$$

(14)

where $\alpha \ge 1$, $\delta > 0$ are constants and

$$p^+(x,t) = \frac{\bar{M}}{\dot{R}^2(t)\bar{b}(x2)} + \left(\propto \frac{T_p}{\sqrt{2}} K\left|\frac{x2}{r(t)}\right| + \delta\right)$$

$$p^-(x,t) = \frac{\bar{M}}{\dot{R}^2(t)\bar{b}(x2)} - \left(\propto \frac{T_p}{\sqrt{2}} K\left|\frac{x2}{r(t)}\right| + \delta\right)$$

(15)

III. Related Work

One of the biggest problems with TCP's congestion control algorithm over drop-tail queues is that sources reduce their transmission rates only after detecting packet loss due to queue overflow. Since a considerable amount of time may elapse between the packet drop at the router and its detection at the source, a large number of numbers of packets may be dropped as the senders continue transmission at a rate the network cannot support.

RED [7] was presented with the objective to minimize packet loss and queuing delay, to avoid global synchronization of sources, to maintain high link utilization, and to remove biases against bursty sources. To achieve these goals, RED utilizes two thresholds, $minth$ and $maxth$, and a exponentially-weighted moving average (EWMA) formula to estimate the average queue length, $Qavg = (1 - Wq) \cdot Qavg + Wq \cdot Q$, where Q is the current queue length and Wq is a weight parameter, $0 \leq Wq \leq 1$. The two thresholds are used to establish three zones. If the average queue length is below the lower threshold ($minth$), RED is in the normal operation zone and all packets are accepted. On the other hand, if it is above the higher threshold ($maxth$), RED is in the congestion control region and all incoming packets are dropped. If the average queue length is between both thresholds, RED is in the congestion avoidance region and the packets are discarded with a certain probability *Pa*:

$$Pa = Pb / (1 - count \cdot Pb)$$

This probability is increased by two factors. A counter is incremented every time a packet arrives at the router and is queued, and reset whenever a packet is dropped. As the counter increases, the dropping probability also increases. In addition, the dropping probability also increases as the average queue length approaches the higher threshold. In implementing this, RED computes an intermediate probability *Pb*,

$$Pb = maxp / maxth - minth \cdot (Qavg - minth)$$

whose maximal value given by $maxp$ is reached when the average queue length is equal to $maxth$. For a constant average queue length, all incoming packets have the same probability to get dropped. As a result, RED drops packets in proportion to the connections' share of the bandwidth.

Unfortunately, when a large number of TCP sources are active, the aggregate traffic generated is extremely bursty [7]. Bursty traffic often defeats the active queue management techniques used by RED since queue lengths grow and shrink rapidly. While ECN [4][11] is necessary for eliminating packet loss in the Internet, we show that RED, even when used in conjunction with ECN, is ineffective in preventing packet loss.

Random Exponential Marking (REM): REM is a framework for communicating congestion information from links to sources by exponential marking. A complete implementation consists of REM marking at the link and REM decoding at the source, however operation on the current Internet with only the link algorithm and TCP has been suggested [8].

Three different alternative pricing algorithms PC1-PC3 constitute REM. PC3 is evaluated here Eq (16), because of its superiority over PC1 and PC2 [8]:

$$PC3:\; p_1(t+1) = [p_1(t) + \gamma(\alpha\, b_1(t) + x^1(t) - c_1)]^+ \quad (16)$$

where x^1 is the aggregate arrival rate, c_1 link capacity, b_1 backlog, and α and γ are pricing constants. An exponential function determines the marking probability from the link price:

$$m = 1 - \varphi^{-p_1(t)}$$

where φ controls marking. Although REM works very well in a steady state situation, the behavior in transient conditions and with realistically constrained buffer sizes is not necessarily optimal. The experimental results detail how in an environment with a wide variation in *N* and finite buffers the performance suffers.

AQM schemes based on feedback control theory: The dynamic models of TCP make it possible to design AQM in the literature of feedback control theory. In [2], RED was considered as a AQM controller whose parameters can be analyzed based on a linearized TCP model. An extension to this work is PI AQM [12], whose marking probability is updated based on the queue length as

$$p(k+1) = p(k) + a\big(q(k+1) - q_{ref}\big) - b(q(k) - q_{ref})$$

where a and b are constants.

It has been shown in [12] that the PI AQM scheme can outperform RED in terms of system response and steady-state error. On the other hand, the PI controller has some inherent limitations: 1) the linearization inevitably introduces model error; 2) it is based on the frequency domain analysis which is invalid for time-varying systems (TCP dynamics are essentially time varying); 3) although gain-phase margin can be analyzed for the PI controller, it can not incorporate robustness directly in the design.

IV. Simulations and Results

In order to evaluate the performance of VS AQM to handle time delays, a number of simulation experiments were run using ns-2 [10] over a small network shown in Fig.2 with varying number of input nodes. To validate the performance and the robustness of the proposed AQM, we conduct a simulation study in different scenarios. Some representative AQM schemes, namely, RED [7], REM [8] and PI [12], are also simulated for the purpose of comparison.

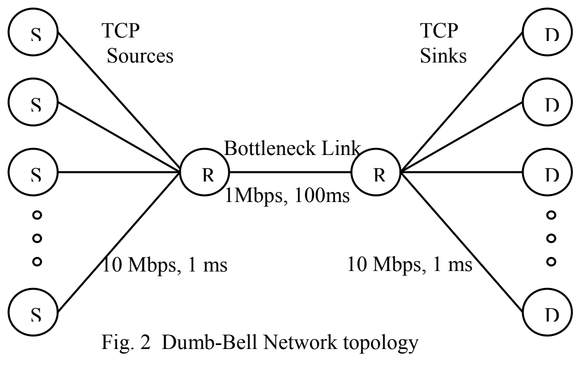

Fig. 2 Dumb-Bell Network topology

A. Simulation Configuration

The dynamic behaviors of the previous AQM schemes are simulated under a variety of network topologies and traffic sources. In particular, we consider the dumb-bell network topology depicted in Fig. 1, where TCP connections share a single bottleneck link. We assume that the TCP sources always have data to send. The links between the TCP sources and the router are 10 Mbps links with a 1 ms propagation delay, which are the same as those between the TCP sinks and the router. Router is connected to through a 1 Mbps 100 ms delay link. The maximum buffer size of each router is set to 300 packets. The packet size was 1040bytes.

The parameters used in VS based on time delays AQM are: $\overline{M} = 100, T_p = 0.2, K = 10, \propto = 1, and\ \delta = 0.3.$

In PI controller, we use the parameter values $a = 1.822\ X\ 10^{-5},\ b = 1.816\ X\ 10^{-5}$. The desired queue length is set to VS based on time delays, PI, RED and REM. The parameters of RED are set as recommended in http://www.aciri.org/floyd/REDparameters.txt.[7]. For REM, the parameters are set as $\alpha = 0.1, \varphi = 1.001, and\ \gamma = 0.001$, which are recommended in [8]. In the following simulations, ECN is enabled for VS AQM based on time delays, PI, REM, and RED and packet loss is observed only when buffer overflows.

B. Scenario of Single Bottleneck Topology

1) Performance Comparison of Different AQM Schemes: In this experiment, we choose $\bar{M} = 100$ in Fig. 2, which corresponds to 100 greedy FTP flows sharing the bottleneck link. The system response using the VS based on time delays AQM is depicted in Fig. 3, where the performance shows fast response and the stabilized queue size. Meanwhile, we repeat the same experiment using PI, RED, and REM, respectively, and depict their instantaneous queue size in Fig. 4.

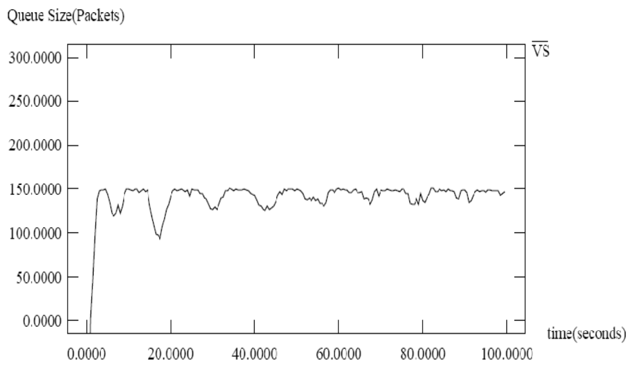

Fig. 3 System Response and Queue Size (in Packets) for VS AQM based on time delays

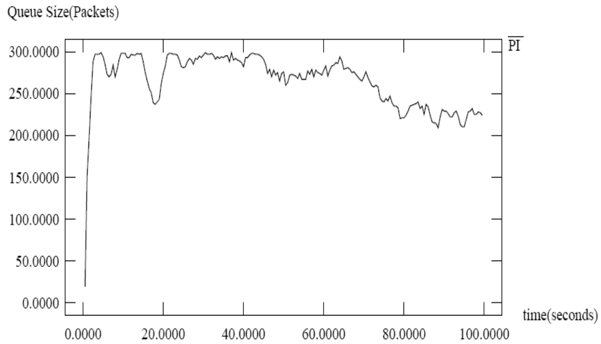

Fig. 4(a) System Response and Queue Size (in Packets) for PI AQM

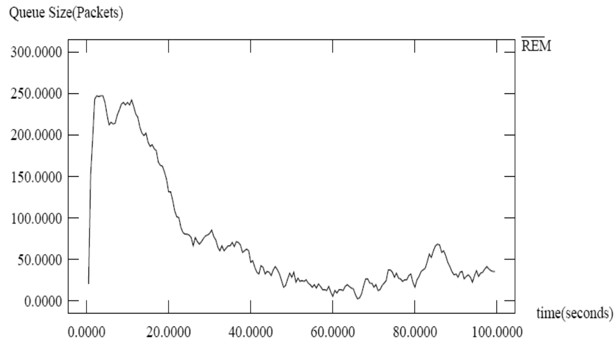

Fig. 4 (b) System Response and Queue Size (in Packets) for REM AQM

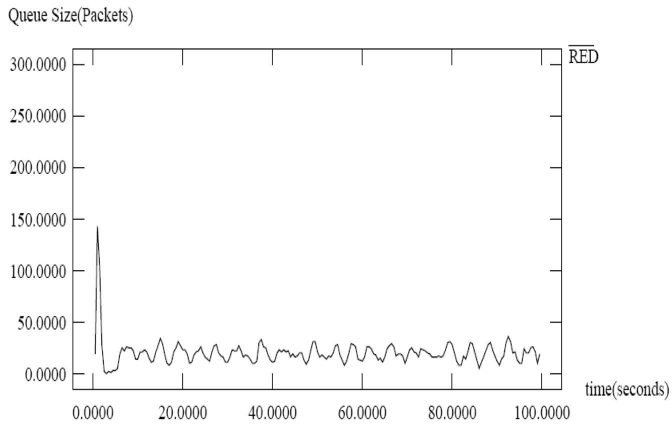

Fig. 4 (c) System Response and Queue Size (in Packets) for RED AQM

As compared to the AQM schemes shown in Fig. 3, we clearly see that the VS-based on time delays AQM outperforms other schemes in terms of system stability and performance, which implies that VS based on time delays AQM in turn have higher link utilization, lower packet loss ratio and smaller queue fluctuation.

2) Performance Under Dynamic Traffic Changes: In this scenario, we provide some time-varying dynamics and investigate the performance of the VS based AQM and other representative schemes. We use 200 TCP connections at time t=0. At time t=40, half of the TCP connections stop transmitting data, and at time t=50 they resume transmitting again. The queue evolution is depicted in Fig. 5. Note that PI [Fig. 5(b)], REM [Fig. 5(c)] and RED [Fig. 5(d)] are not very robust with respect to such connection number variation, which result in heavy queue fluctuation during 40–50s. Although RED [Fig. 5(d)] is not very sensitive in this scenario, it tends to over-mark the incoming traffic so that the link utilization is degraded. As is evident from [Fig. 5(a)], the

VS based AQM is very robust against the variation of connections and keeps very good response even in the presence of such variations.

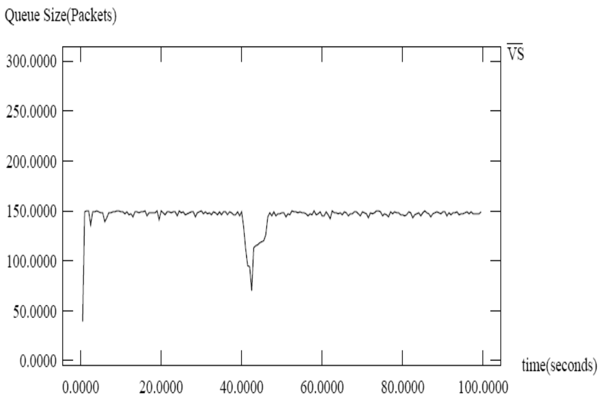

Fig. 5 (a) Queue Evaluation using VS based AQM

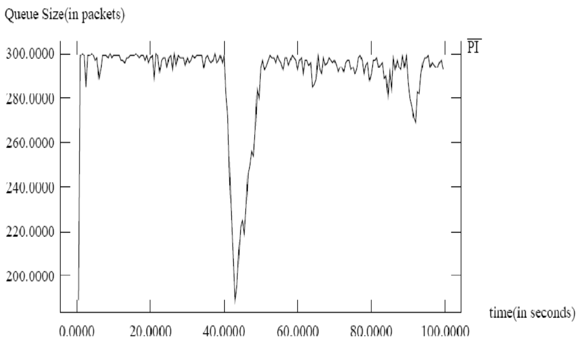

Fig. 5 (b) Queue Evaluation using PI

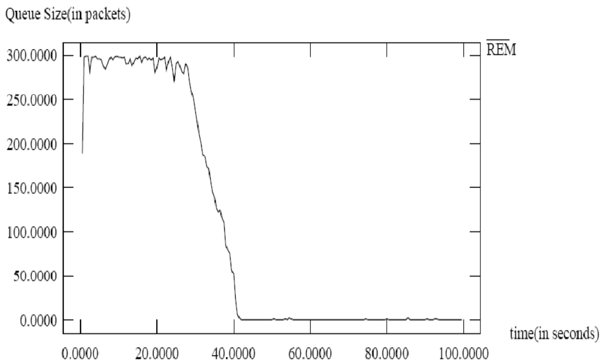

Fig. 5 (c) Queue Evaluation using REM

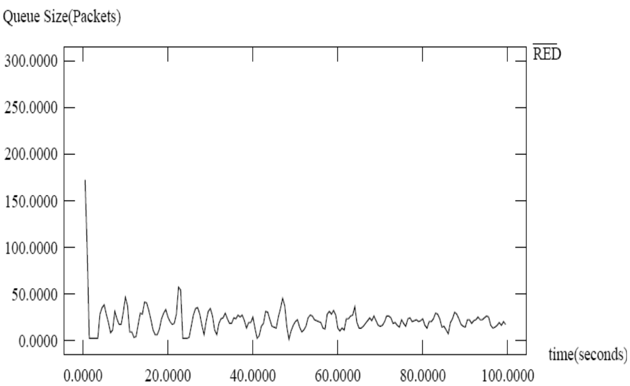

Fig. 5 (d) Queue Evaluation using RED

3) Robustness w.r.t. Number of TCP Connections: The performance and robustness of the VS based AQM are explored with respect to different TCP loads. We conduct simulations with the same setting as in Experiment 1, except that the number of connections varies from 100 to 300. Fig. 6 plots the average queue length (from 0s to 100 s) for different AQM schemes. It is observed that the VS based AQM can robustly stabilize the queue length around 100 packets. The average queue lengths of REM and RED vary with respect to the flow number.

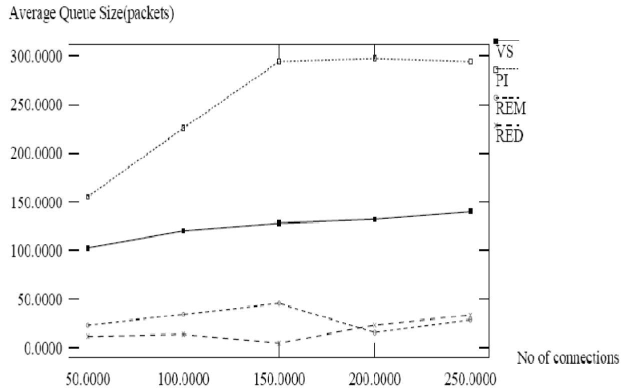

Fig. 6 Average queue length with respect to the number of TCP flows

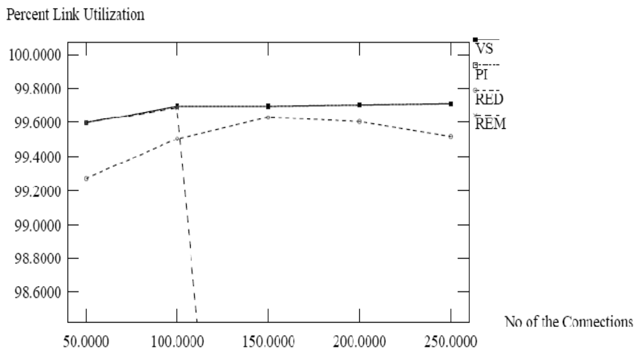

Fig. 7 Percent Link Utilization w.r.to number of TCP flows

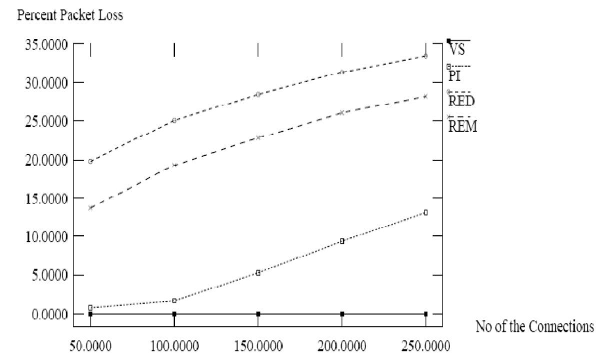

Fig. 8 Percent packet loss w.r.to number of TCP flows

Fig. 7 shows the percentage of link utilization over number of connections, the percentage of link utilization over buffer size, respectively. We clearly see that the VS BASED AQM have better link utilization than REM and RED, and the link utilization of REM and RED are sensitive to the variation of the flow numbers. Fig. 8 shows the percentage of packet loss over number of connection. Compared with PI, RED, and REM, the VS BASED AQM has much better performance in terms of robustly stabilized queue length, high link utilization, stable link utilization and low percentage of packet loss.

4) Comparison with PI, RED and REM AQM under Higher TCP Loads: In this experiment, we set the flow number to

$\bar{M} = 500$ and compare the system response of PI, RED, REM and the VS based AQM. As shown in Fig. 9, the VS BASED AQM has very good transient response and very low overshoot. On the other hand, the RED and REM exhibit much slower response and larger overshoot, which implies higher packet loss ratio and larger RTT.

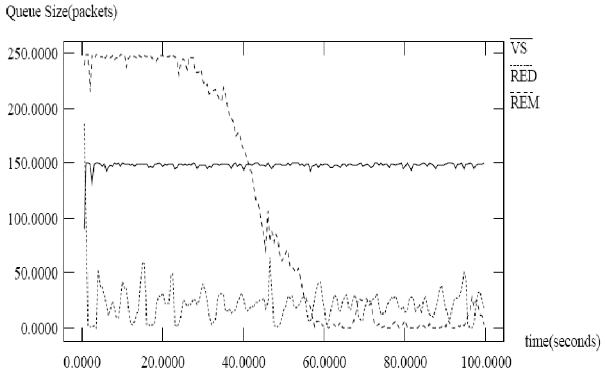

Fig. 9 System Responses under Higher TCP Loads Integral

6) Robustness w.r.t. RTT: The VS BASED AQM has very good robustness against the uncertainty of RTT, which is essential as an AQM scheme. In this experiment, we change the propagation delays in the network topology (Fig. 1), $\bar{M} = 200$ and evaluate the robustness of the VS BASED AQM. First, we set the propagation delay between router R1 and R2 to 4 ms and the delays between the routers and the end hosts 40 ms, which corresponds to a much smaller RTT than the nominal one. The regulated queues are depicted in Fig. 10 where the VS BASED AQM is still capable of robustly stabilizing the queue length.

Meanwhile, we also consider the scenario with a much larger RTT, where we set the propagation delay between R1 and R2 to 300 ms and the delays between the routers and the end hosts 10 ms. We repeat the simulation with the flow number N=200 and obtain the system responses depicted in Fig. 11. As we can observe, the VS BASED AQM continues to exhibit good performance in the sense of queue stability and fast response, which outperforms other peer schemes.

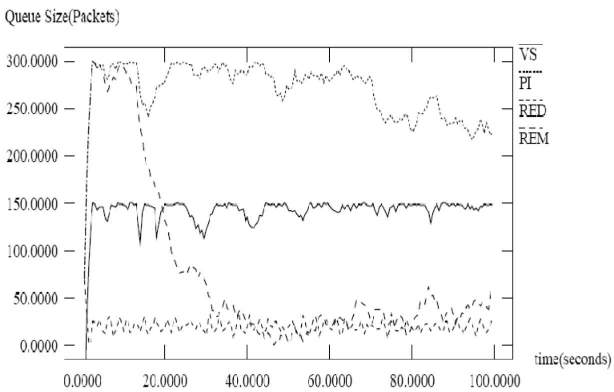

Fig. 10 Instantaneous Queue Size with a smaller propagation delay

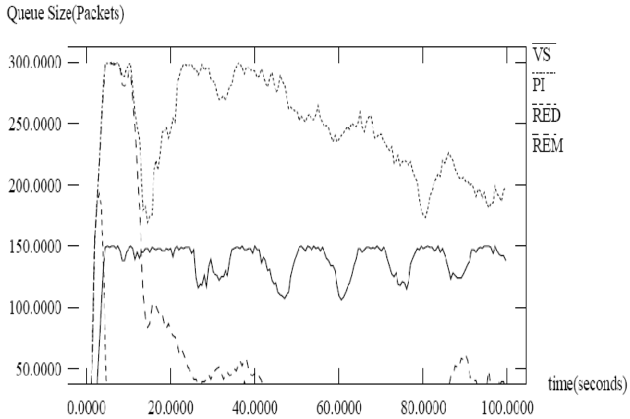

Fig. 11 Instantaneous Queue Size with a larger propagation delay

V. Conclusion

In this paper, we developed a VS based AQM scheme supporting ECN. We presented guidelines for designing the robust VS based AQM to handle time delays for the TCP dynamics. The robustness with respect to the Round Trip Times and the number of the active TCP connections was analyzed, and the different performance metrics was discussed. It was discovered that the VS based AQM has many desirable properties such as good robustness and fast system response. We also provided *ns* simulations in different scenarios to validate our results. The simulation experiments showed that the planned AQM scheme performs better than a number of AQM schemes in terms of packet loss ratio, link utilization and queue fluctuation. This work will be extended into Differentiated Services Network also, which we are currently investigating.

References

[1] R. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L.Peterson, K. Ramakrishnan, S.Shenker, J. Wroclawski, and L. Zhang, “Recommendations on queue management and congestion avoidance in the Internet”, RFC 2309, Apr 1998.

[2] C. Hollot, V. Misra, D. Towsley, and W. Gong, “A control theoretic analysis of RED,” in *Proc. IEEE INFOCOM*, Alaska, Apr. 2001, pp. 1510–1519.

[3] S. Floyd, “TCP and explicit congestion notification”, Comput. Commun. Rev., vol. 24, no. 5, pp. 10-23, Oct. 1994.

[4] K. Ramakrishnan and S. Floyd, “A proposal to add Explicit Congestion Notification (ECN) to IP”, RFC 2481, Jan. 1999. July 1993.

[5] V.Jacobson,“Congestion avoidance and control”, in Proc. ACM SIGCOMM, Aug 1998, pp. 314-329.

[6] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the Internet”, IEEE/ACM Transaction on Networking, vol. 7, pp, 458-472, August 99.

[7] S.Floyd and V. Jacobson, “Random early detection for congestion avoidance”, IEEE/ACM Transactions Networking, Vol. 1, pp 397-413,

[8] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, “REM: Active queue management,” *IEEE Network*, vol. 15, no. 3, pp. 48–53, May-Jun. 2001.

[9] Peng Yan, Yuan Gao, Hitay Ozbay, “A Variable Structure Control Approach to Active Queue Management for TCP with ECN”, IEEE Transactions on control systems technology, Vol.13, No.2, PP 203-215, March 2005.

[10] The network simulator ns-2 (2000). [Online]. Available: http://www.isi.edu/nsnam/ns/

[11] K. Ramakrishnan, S. Floyd and D. Black, “The addition of Explicit Congestion Notification (ECN) to IP”, RFC 3168, Sep. 2001.

[12] _________ , “Analysis and design of controllers for AQM routers supporting TCP flows,” *IEEE Trans. Autom. Control*, vol. 47, no. 6, pp. 45–959, Jun. 2002.

[13] V. Misra,W. Gong, and D. Towsley, “A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to

RED," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Sep. 2000, pp. 151–160.
[14] F.Kelly, Mathematics Unlimited—2001 and Beyond, B. Engquist andW. Schmid, Eds. Berlin, Germany: Springer-Verlag, 2001. Mathematical modeling of the Internet.
[15] S. Low, F. Paganini, J. Wang, S. Adlakha, and J. Doyle, "Dynamics of TCP/RED and a scalable control," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, pp. 239–248.
[16] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electromechanical Systems*. New York: Taylor & Francis, 1999.

Dr.A.M.Natarajan is the CEO & Professor at Bannari Amman Institute of Technology, Sathyamangalam, Coimbatore. He received the B.E. degree in Electrical Engineering, and M.Sc.(Engg.) in Applied Electronics and Servo Mechanism from Madras University and Ph.D degree in System Engineering from Madras University. He has 40 years of teaching experience. He received the "Best Engineering college principal award in India for the Year 2000**"** from Indian Society for Technical Education, New Delhi. He has published more than 70 papers in International and National journals and conference proceedings. He has published 10 books. His areas of research include systems engineering and computer networks.

V. Santhi received the B.E. degree in Computer Science and Engineering from Madras University with distinction. She received M.E. degree in Computer Science and Engineering from Anna University Chennai with distinction. She has 7 years of teaching experience. She is currently pursuing doctoral research in computer networks management. At present she is working as Lecturer in Computer Science and Engineering Department at PSG College of Technology, Coimbatore.