Analysis and Design of AQM based on State-Space Models for Stabilizing TCP [1]

Ki Baek Kim and Steven H. Low

†Departments of CS and EE
California Institute of Technology, Pasadena, CA 91125
kkb@cisl.snu.ac.kr, slow@caltech.edu

Abstract

In this paper, we formulate the AQM (Active Queue Management) design problem for stabilizing a given TCP (Transmission Control Protocol) as state-space models. Thereby, we show that PD-type (Proportional-Derivative) AQM is a natural state-feedback control structure to stabilize the given TCP for the first time in the networking literature to our knowledge and by applying integral control action, a PID-type (Proportional-Integral-Derivative) control structure that is a unified framework for analysis and design of AQM. Second, we compensate for delays in congestion measure explicitly by using a memory control. Third, we obtain stabilizing optimal gains of the proposed feedback control structures for linearized systems of the given TCP. We interpret existing AQMs, including a simplified RED (Random Early Detection), REM (Random Exponential Marking), PI (Proportional-Integral) and a simplified AVQ (Adaptive Virtual Queue), as different approximations of the unified AQM structure. We discuss the impact of each structure on performance from the results of the stabilizing optimal AQMs. Finally, we illustrate our results through ns simulations for TCP Reno.

1 Introduction

Congestion control is a distributed iterative procedure to maximize utilization of given network resources and share them efficiently among competing sources. It consists of local algorithms executed dynamically at sources (TCP) and at links (AQM). Links update, implicitly or explicitly, a measure of congestion, and feed it back to sources by dropping or marking arrival packets. In response, sources adjust their rates based on the feedback information from links in their paths. Popular TCP algorithms include Reno (and its variants) [3] and Vegas [4], and popular AQM algorithms include DropTail, RED and its variants [5].

There are many ways to formulate congestion control problems. The papers [6, 7] have formulated the internet control problem as a primal-dual model. The papers [8, 9] have expressed TCP as a dynamical system consisting of sources, queues, and feedback control from AQM, and AQM as a control action, where the queue itself is a dynamical system. The papers [8, 9, 10] then formulate the AQM design problem for the given TCP as a transfer function model. In general, the equilibrium points of a TCP/AQM pair and dynamic behavior depend on both the TCP/AQM algorithms and the queue dynamics. The primal-dual approach in [6, 7] provides a unified framework to compare different TCP algorithms in terms of equilibrium allocation (i.e., fairness). However, a similar model is lacking to compare and understand various AQMs in terms of structures and performance. In addition, existing AQMs do not address what is a natural feedback control to stabilize the given AQM, systematically. As a result, AQM algorithms are only compared in the literature through simulations. Therefore, it is important to develop a mathematical framework for analysis and design of AQMs.

To our knowledge, existing AQM algorithms only use the current dynamic information and thus do not compensate for large delays explicitly. Since they only use the current dynamic information, they will have difficulty to stabilize the given TCP algorithm and thus get maximum throughput in the presence of large delays. Thus, we need to investigate how to compensate for large delays explicitly.

In this paper, we formulate the AQM design problem for stabilizing a given TCP as state-space models. Thereby, we show that PD-type AQM structure is a natural state-feedback control structure to stabilize the given TCP for the first time in the networking literature to our knowledge, which the primal-dual models in [6, 7, 11, 12] and existing AQM algorithms in [5, 9, 13, 14] did not capture. By applying integral control action, we obtain a PID-type control structure that is a unified mathematical framework for analysis and design of AQM. Second, we compensate for delays in the feedback control explicitly by using a memory control structure. Third, we get stabilizing optimal gains of the proposed AQM structures for linearized systems of the given TCP. We interpret existing AQMs including a sim-

---

plified RED, REM/PI, and a simplified AVQ, as different approximations of the unified AQM structure. We discuss the impact of each structure on performance from the results of the stabilizing optimal AQMs. Finally, we illustrate the proposed results via *ns* simulations.

Although we consider TCP Reno as an exercise for the analysis and design of AQM, the proposed procedures and results in this paper can be applied directly to the AQM design problem for any TCP algorithms. Also, although we consider the AQM design problem at routers in order to support TCP algorithms, the derived feedback control structures in this paper can be implemented at sources (end-users). As a first step to develop a unified mathematical framework of AQM, we consider stability of the linearized system as in [5, 9, 13, 14]. However, we verify our results using the well-known *ns* simulator that has stochastic and nonlinear dynamics of the network.

All proofs of theorems, together with many other details, are omitted due to the page limit and can be found in [1, 2].

2 PD-type and PID-type AQM structures

Consider the simple case of a single link of capacity c shared by N TCP Reno sources, modeled by

$$\dot{w}_s(t) = \frac{w_s(t - \tau_s(t))}{d_s + b(t - \tau_s(t))/c}(1 - p(t - \tau^b(t)))\frac{1}{w_s(t)}$$
$$- \frac{w_s(t - \tau_s(t))}{d_s + b(t - \tau_s(t))/c}p(t - \tau^b(t))\frac{w_s(t)}{2}$$
$$\dot{b}(t) = -c + \sum_{s=1}^{N} \frac{w_s(t - \tau_s^f(t))}{d_s + b(t - \tau_s^f(t))/c}$$
$$= -c + y(t - \tau^f(t)), \qquad (1)$$

where w_s is TCP window size of source s, τ_s is the round trip time, τ^b is the backward delay from link to source, τ^f is the forward delay from source to link, p is the loss probability, b is the queue length, y is aggregate rate, and c is the link capacity in packets/sec. We define the source rate by $x_s(t) = w_s(t)/(d_s + b(t)/c)$, where d_s is the round trip propagation delay of source s. As in [9], we assume sources are identical $d_s = d$ and all have the common window $w_s(t) \equiv w(t)$; we assume delays take their equilibrium values and are constant, and forward delays are zero, $\tau_s^f(\cdot) = 0$, so that $\tau_s(t) = \tau_s^b(t) = \tau$. Let (w^*, b^*, p^*) be the equilibrium point. Then τ is related to b^* by $\tau = d + b^*/c$.

The first key step to proposing a new AQM structure is to convert the system (1) to the equivalent form

$$\ddot{b}(t) = f(b(t), \dot{b}(t), \dot{b}(t - \tau), p(t - \tau)). \qquad (2)$$

Then, we can derive the following state-space model of the linearized TCP and queue dynamics:

$$\dot{z}(t) = Az(t) + B\delta p(t - \tau), \qquad (3)$$

where $z(0)$ and $\{\delta p(\sigma), \sigma \in [-\tau, 0]\}$ are given,

$$z(t) = \begin{bmatrix} \delta b(t) \\ \delta \dot{b}(t) \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ A_1 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ B_1 \end{bmatrix}$$

$$A_1 = -\frac{2cN}{\tau(2N^2 + c^2\tau^2)}, \quad B_1 = -\frac{2N^2 + c^2\tau^2}{2\tau^2 N}$$

$$A_2 = -\frac{2cN\tau + 2N^2 + c^2\tau^2}{\tau(2N^2 + c^2\tau^2)}. \qquad (4)$$

Since $A_2^2 + 4A_1 > 0$ for system matrices of (5), we have $a_1 \neq a_2 \neq 0$, $e_1 < 0$ and $B_1 < 0$ for $\tau > 0$. Note that $a_1 = a_2$ when $2N = \tau c$ even for $\tau > 0$ in the linearized model of [15].

From the above state-space model, we can naturally get a PD-type state-feedback AQM structure $(\delta p(t) = Kz(t) = K_P \delta b(t) + K_D \delta \dot{b}(t))$ if we use only the current dynamic information, where RED is a P-type control.

Remark 1 *It is interesting to see that if we ignore the delay term, the transfer function from $\delta b(t)$ to $\delta w(t)$ is equal to the lead-lag compensator with the form*

$$\frac{\delta w(s)}{\delta b(s)} = \frac{k_2 s + k_3}{s + k_1},$$

where k_1, k_2, and k_3 are some constants, and $\delta w(t)$ is a variation of $w(t)$ near the equilibrium point. Here, we emphasize that the primal-dual models in [6, 7, 11, 12] and existing AQM algorithms in [5, 9, 13, 14] did not capture the derived PD-type feedback control structure. We also emphasize that the proposed PD-type structure in this paper can also be implemented either at sources or at routers with congestion measures different from the buffer $\delta b(t)$, where the feedback control structures in Vegas [4] and a new static TCP [12] can be considered as P-type controls in terms of queue length. Note that [16, 17] and [18, 19] add D-type control to [4] and [12], respectively after our papers [1, 2] have shown that PD-type control is a natural state-feedback structure to stabilize the given TCP.

Next, we propose a unified mathematical framework for analysis and design of AQM by applying integral control action technique that augments the original system by differentiating the control in order to make the steady-state tracking-error go to zero for linear systems [20]. *The key step to applying the technique is to have another derivative of the system (1) as follows* $\dddot{b}(t) = g(b(t), \dot{b}(t), \ddot{b}(t), b(t - \tau), \ddot{b}(t - \tau), p(t - \tau), \dot{p}(t - \tau))$.

Then, we can derive the third order TCP model:

$$\dot{z}_e(t) = A_e z_e(t) + B_e \delta \dot{p}(t - \tau), \qquad (5)$$

where $z_e(0)$ and $\{\delta \dot{p}(\sigma), \sigma \in [-\tau, 0]\}$ are given,

$$z_e(t) = \begin{bmatrix} z_0(t) \\ z(t) \end{bmatrix}, \quad A_e = \begin{bmatrix} 0 & I_e \\ 0 & A \end{bmatrix}, \quad B_e = \begin{bmatrix} 0 \\ B \end{bmatrix}$$

with $z_0(t) = I_e z(t)$ and $I_e = [1, 0]$.

From the above state-space model, we can naturally get a PID-type state-feedback AQM structure $(\delta \dot{p}(t) = Kz(t) = K_I \delta b(t) + K_P \delta \dot{b}(t) + K_D \delta \ddot{b}(t))$ if we use only the current dynamic information. This kind of control is called a memoryless control in the literature.

If we use a memoryless control, it is well known in the control literature that we have performance limit in the presence of large delays τ, i.e., we cannot get good throughput and stabilize the TCP and queue dynamics in the presence of large delays τ. We overcome this problem in the next section.

From the next section, we consider only PID-type AQM structure since we can handle PD-type AQM structure as a special case of PID-type AQM structure.[1]

3 Explicit delay compensation

In order to derive a memory control explicitly for the delayed system, we define

$$e_1 = e^{-a_1\tau} - e^{-a_2\tau}, \quad e_2 = a_1 e^{-a_1\tau} - a_2 e^{-a_2\tau}$$
$$e_3 = a_2 e^{-a_1\tau} - a_1 e^{-a_2\tau}$$
$$e_4 = \frac{-1}{e_1}\left[\frac{a_1}{a_2}e^{-a_1\tau} - \frac{a_2}{a_1}e^{-a_2\tau} + \frac{a_2^2 - a_1^2}{a_1 a_2}e^{-A_2\tau}\right]$$
$$e_5 = \frac{1}{e_1}\left[\frac{1}{a_2}e^{-a_1\tau} - \frac{1}{a_1}e^{-a_2\tau} + \frac{a_2 - a_1}{a_1 a_2}e^{-A_2\tau}\right]$$
$$a_1 = \frac{A_2 + \sqrt{A_2^2 + 4A_1}}{2}, \quad a_2 = \frac{A_2 - \sqrt{A_2^2 + 4A_1}}{2}$$
$$\hat{B}_1 = \frac{B_1(a_2 - a_1)e^{-A_2\tau}}{e_1}. \tag{6}$$

The key to deriving explicit memory controls for the delayed system (5) is to transform the delayed system (5) to the equivalent nominal system

$$\dot{s}_e(t) = A_e s_e(t) + \hat{B}_e \delta \dot{p}(t), \tag{7}$$

where

$$s_e(t) = [s_1, s_2, s_3]^T, \quad \hat{B}_e = [0, \hat{B}^T]^T$$
$$s_1(t) = \frac{(a_2 - a_1)e^{-A_2\tau}}{e_1}(\delta b(t) + u_{1\tau}(t)) +$$
$$\qquad e_4(\delta \dot{b}(t) + u_{2\tau}(t)) + e_5(\delta \ddot{b}(t) + u_{3\tau}(t)) \tag{8}$$
$$s_2(t) = -\frac{e_2}{e_1}(\delta b(t) + u_{2\tau}(t)) + \delta \dot{b}(t) + u_{3\tau}(t) \tag{9}$$
$$s_3(t) = A_1(\delta b(t) + u_{2\tau}(t)) + \frac{e_3}{e_1}(\delta \ddot{b}(t) + u_{3\tau}(t)) \tag{10}$$

---

[1]Inner and outer loop PI-type AQM to the aggregate rate in [10] can be represented as a memoryless PID-type AQM. Note that the PID-type AQM in this paper is reported in [1, 2] before the paper [10] is published.

$$\begin{bmatrix} u_{1\tau}(t) \\ u_{2\tau}(t) \\ u_{3\tau}(t) \end{bmatrix} = \int_{-\tau}^{0} \begin{bmatrix} \frac{a_1 - a_2}{a_1 a_2} + \frac{e^{-(\sigma+\tau)a_1}}{a_1} - \frac{e^{-(\sigma+\tau)a_2}}{a_2} \\ e^{-(\sigma+\tau)a_1} - e^{-(\sigma+\tau)a_2} \\ a_1 e^{-(\sigma+\tau)a_1} - a_2 e^{-(\sigma+\tau)a_2} \end{bmatrix}$$
$$\delta \dot{p}(\sigma + t)d\sigma \left(\frac{B_1}{a_1 - a_2}\right). \tag{11}$$

Thus, by adding the memory control $u_{i\tau}(t)$, we can remove the delay in congestion measure from the original linearized system.

Note that the closed-loop system of (7) is asymptotically stable if and only if the transformed system (5) is asymptotically stable.

4 Stabilizing Optimal AQMs: RHAs

In this section, we obtain optimal gains of the proposed control structures. As a performance measure, consider

$$\min_{\delta \dot{p}(\cdot)} J = \int_{t}^{t+\infty} \left[s_e^T(\sigma)Qs_e(\sigma) + \delta \dot{p}^2(\sigma)\right]d\sigma, \tag{12}$$

where the pair $(A_e, Q^{\frac{1}{2}})$ is observable.

Theorem 1 *The stabilizing optimal AQM, which is called RHA in this paper, for the system (7) is given by*

$$\delta \dot{p}^*(t) = H_1^h(\delta b(t) + u_{1\tau}(t)) + H_2^h(\delta \dot{b}(t) + u_{2\tau}(t))$$
$$\qquad + H_3^h(\delta \ddot{b}(t) + u_{3\tau}(t)). \tag{13}$$

Theorem 1 clarifies the structure of the optimal AQM that stabilizes TCP dynamics (7) at the minimum cost as defined in (12). It implies in particular that the computation of the marking probability should be based on the perturbations in queue length $(\delta b(t) = b(t) - b^*)$, in aggregate rate $(\delta \dot{b}(t) = \delta y(t))$, in the rate of change in aggregate rate $(\delta \ddot{b}(t) = \delta \dot{y}(t))$, and delay compensation $(u_{i\tau}(t))$. Intuitively, excess queue and aggregate rate should lead to an increase in marking probability, and hence the dependence on $\delta b(t)$ and $\delta \dot{b}(t)$.

Because of the delay in congestion measures, the control should correct for the error in input over the previous delay period. This correction is represented by the extra terms $(u_{i\tau}(t))$. We discuss in the end of this section the effect of H_i on the system behavior.

Theorem 1 implies that the solution of the problem (12) is an AQM algorithm, specified by (H_1, H_2, H_3). Conversely, given any AQM of this structure, it solves problem (12) with appropriate weights Q_i, as the next result says. It can be easily proved from Theorem 1.

Theorem 2 *Given a stabilizing AQM* $\delta \dot{p}(t) = [H_1 \ H_2 \ H_3] s_e(t)$, *it solves problem (12) with weights*

$$Q_1 = H_1^2, \quad Q_3 = H_3^2 + 2\frac{A_2 H_3 + H_2}{\hat{B}_1}$$

$$Q_2 = H_2^2 - 2\frac{A_2 H_1 + \hat{B}_1 H_1 H_3 - A_1 H_2}{\hat{B}_1}. \quad (14)$$

In order for an AQM $\delta \dot{p}(t) = [H_1 \ H_2 \ H_3] s_e(t)$ to be a stabilizing optimal control when the system is controllable and observable for the system (5), it should satisfy $H_1 > 0$, $H_2 > -\frac{A_1}{\hat{B}_1}$, and $H_3 > 0$.

Alternatively, an AQM can be specified by the eigenvalues of the desired closed-loop system.

Theorem 3 *Given the eigenvalues* λ_1, λ_2, *and* λ_3 *of the closed-loop system (7) with (13),* $\delta \dot{p}(t)$ *solves problem (12) with weights*

$$Q_1 = \frac{\hat{\lambda}_3^2}{\hat{B}_1^2}, \quad Q_2 = \frac{-A_1^2 + \hat{\lambda}_2^2 - 2\hat{\lambda}_1 \hat{\lambda}_3}{\hat{B}_1^2}$$

$$Q_3 = \frac{-A_2^2 - 2A_1 + \hat{\lambda}_1^2 - 2\hat{\lambda}_2}{\hat{B}_1^2}, \quad (15)$$

where $\hat{\lambda}_1 = \lambda_1 + \lambda_2 + \lambda_3$, $\hat{\lambda}_2 = \lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_1 \lambda_3$, $\hat{\lambda}_3 = \lambda_1 \lambda_2 \lambda_3$.

4.1 Approximating AQMs

We now interpret AQM proposals a simplified RED without a low-pass filter, REM, PI and a simplified AVQ without a virtual queue dynamics as various approximations of unified AQM structure in this paper. Since those existing AQMs do not use the memory control, for easy comparison, we assume that $\tau = 0$ (i.e., $s(t) = z(t)$, $\hat{B} = B$, $s_e(t) = z_e(t)$, and $\hat{B}_e = B_e$) for linearized model. The models we use for these schemes are slightly simplified.

Then, the linear models motivated by these AQMs are:

simplified RED: $\delta \dot{p}^r(t) = H_2^r \delta b(t)$

simplified AVQ: $\delta \dot{p}^a(t) = H_2^a \delta b(t)$ (16)

REM/PI: $\delta \dot{p}^m(t) = H_1^m \delta b(t) + H_2^m \delta b(t)$

for some nonnegative constants $H_2^r, H_2^a, H_1^m, H_2^m$.

By Theorem 1, the stabilizing optimal AQM has a strictly positive gain $H_3 > 0$. Since this condition is satisfied by none of RED, REM, PI and AVQ, none of them can be made optimal, in the sense of minimizing (12), by tuning its parameters. Moreover, their structure implies a limitation to rate of convergence to equilibrium.[2]

---
[2] We caution however that the analysis applies only when the linear models we use are reasonable approximations of these AQMs. But, the proposed design procedures can be applied to any kind of AQMs including RED with a low-pass filter and AVQ with a virtual queue dynamics.

Specifically, RED (AVQ) have H_1^r and $H_3^r = 0$ ($H_1^a = 0$ and $H_3^a = 0$). From Theorem 2, the sum of eigenvalues of the closed-loop system is given by

$$\lambda_1 + \lambda_2 + \lambda_3 = A_2 + B_1 H_3^r \leq A_2,$$

where the last inequality follows from that $A_2 < 0$, $B_1 < 0$, and $H_3^r \geq 0$. Since all eigenvalues have nonpositive real parts, the above inequality means that the sum of the real parts of the eigenvalues is less negative when $H_3^r = 0$ than when $H_3^r > 0$. This suggests that the decay rate is smaller with RED ($H_3^r = 0$). The implication of $H_1^r = 0$ is that at least one of the eigenvalues λ_i is zero, implying that the convergence rate is very slow and the original nonlinear system could be unstable according to the center manifold theorem [21].

Since $H_3^m = 0$ for REM and PI, they suffer from the same structural limitation on decay rate and stability as RED. This is illustrated in the simulation below.

As shown in [1], the cost of RED, AVQ, and REM/PI can be obtained from $J^* = z_e^T(0) K z_e(0)$ by setting some elements of H_i to zero ($H_1 = H_3 = 0$ for RED/AVQ and $H_3 = 0$ for REM/PI). Note that the costs of RED, REM, and AVQ are always greater than that of the RHA for the same weighting matrices Q_i.

5 *ns* Simulations for TCP Reno

In this section, we illustrate the proposed results via two simulation scenarios.

For ease of implementation of stabilizing optimal AQMs, we set all eigenvalues of the closed-loop system to be equal (i.e., $\lambda = \lambda_i = -3.5$ for all i). Then, we have only to design $\lambda < 0$. The other values of stabilizing optimal AQMs are obtained automatically from the equations of Section 4. For detailed implementation, refer to [1].

P-, PI-, and PD- type AQMs are obtained by setting $H_1 = H_3 = 0$, $H_3 = 0$, and $H_1 = 0$ after getting gains H_1, H_2, and H_3 of PID-type AQM.

We simulate a single bottleneck link with capacity $c = 4000$ pkts/sec shared by $N = 100$ Reno sources. The AQM uses ECN marking. Queue length is limited by $b_{max} = 800$ pkts. Then, the target queue length is set to $b^* = 100$ pkts. Marking probability of each AQM is updated every 2ms, i.e., the sampling time is 2ms. The simulation duration is 30 sec, where this duration is enough to illustrate the proposed results.

5.1 Comparison of memoryless AQMs

Here, we compare performances of P-, PEM/PI-, PD-, and PID- type memoryless AQMs without explicit delay compensation in the presence of delays. Assume that $d_s = 150$ms, where $\tau^* = d_s + b^*/c$.

Figure 1 shows performances of each AQM. I-type control can make the queue length large at the transient-state, while it makes the queue length go to the target queue length fast at the steady-state. Conversely, D-type control can make the queue length approach the target queue length fast at the transient-state, but cannot make the steady-state tracking error between the target and real queue lengths approach zero. The mean queue lengths of P-, PD- and PID- type are 78.96, 77.95, and 99.4793, respectively. Thus, the simulation results tell us that PID-type AQM is more stabilizing than the other type AQM structures without hurting throughput and without increasing queuing delay. All of them are stabilizing and thus have 96 ∽ 97 % utilization. However, if AQM cannot make TCP stable in the presence of large delays and capacities, the throughput will decrease drastically.

5.2 Performance of a memory control

Here, we illustrate performance of a memory control by comparing a memory RHA with a memoryless RHA.

Figures 2 and 3 show queue trajectories of PID-type memoryless AQM and PID-type memory AQM, respectively when $d_s = 150$msec, 250msec and 350msec. Refer to figure 1 for performance of PID-type memoryless AQM when $d_s = 150$msec. They show that PID-type memoryless AQM becomes unstable as the propagation delay increases, while PID-type memory AQM can compensate large delays appropriately. Utilization of PID-type memory AQM is higher than that of PID-type memoryless AQM in the presence of large delays as shown in Table 1. Thus, the simulation results tell us that we need a memory control in order not only to stabilize TCP and but also to maximize throughput in the presence of large delays.

6 Conclusion

In this paper, we formulated the AQM design problem as state-space models. Thereby, we showed that PD-type and PID-type control structures are necessary to maximize throughput as well as stabilize the given TCP, proposed one way to compensate for delays in congestion measure explicitly, and obtained stabilizing optimal gains of the proposed AQM structures. We interpreted existing AQMs as different approximations of the proposed unified AQM structure. We discussed the impact of each structure on performance. Finally, we illustrated the proposed results via ns simulations.

This proposes what kind of state-feedback control structure is necessary to stabilize the given TCP, where it clarifies that P-type control RED is not enough to stabilize the given TCP model in [9]. Second, it proposes one way to compensate for delays in control explicitly. Although we consider the AQM design problem at routers in order to support TCP algorithms, the derived control structures in this paper can also be implemented at sources (end-users), where feedback controls in Vegas [4] and a new static TCP [12] can be con-

sidered as P-type controls in terms of queue length. Because of these two important features, we expect that the proposed results will be useful for the analysis and the design of other dynamical systems with similar structure.

References

[1]   K. B. Kim and S. H. Low, "Analysis and design of AQM for stabilizing TCP." Caltech Technical Report caltechCSTR:2002.009, March 2002. http://cisl.snu.ac.kr/~kkb, http://caltechcstr.library.caltech.edu/, http://netlab.caltech.edu.

[2]   K. B. Kim and S. H. Low, "Design of receding horizon AQM in stabilizing TCP with multiple links and heterogeneous delays," in *Proc. of 4th Asian Control Conference*, vol. WA-1, (Singapore), 2002. http://cisl.snu.ac.kr/~kkb.

[3]   V. Jacobson, "Congestion avoidance and control," in *Proceedings of ACM/SIGCOMM*, 1988. ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z.

[4]   L. S. Brakmo and L. L. Peterson, "TCP vegas: end to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, 1995. http://netweb.usc.edu/yaxu/Vegas/Reference/brakmo.ps.

[5]   S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993. ftp://ftp.ee.lbl.gov/papers/early.ps.gz.

[6]   F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, pp. 237–252, 1998. http://www.statslab.cam.ac.uk/~frank/mmi.html.

[7]   S. H. Low and D. E. Lapsley, "Optimization flow control, I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 861–874, December 1999. http://netlab.caltech.edu.

[8]   J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Computer Networks*, vol. 36, pp. 203–235, 2001.

[9]   C. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings of IEEE/INFOCOM*, 2001. http://www-net.cs.umass.edu/networks/publications.html.

[10]   H. Lim, K. J. Park, E. C. Park, and C. H. Choi, "Active queue management algorithm with a rate regulator," in *Proc. of 15st IFAC World Congress on Automatic Control*, vol. T-We-M02, (Barcelona, Spain), 2002.

[11] G. Vinnicombe, "A new TCP with guaranteed network stability," tech. rep., Cambridge Univiversity, preprint, June 2001.

[12] S. H. Low, F. Paganini, J. Wang, S. A. Adlakha, and J. C. Doyle, "Dynamics of TCP/AQM and a scalable control," in *Proceedings of IEEE Infocom*, June 2002.

[13] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management," in *Proceedings of ACM/SIGCOMM*, 2001. http://comm.csl.uiuc.edu/~srikant/pub.html.

[14] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: active queue management," *Submitted for publication*, 2000. http://netlab.caltech.edu/pub.html.

[15] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A control theoretic analysis of RED," in *Proceedings of IEEE/INFOCOM*, April 2001. http://www-net.cs.umass.edu/papers/papers.html.

[16] D. H. Choe and S. H. Low, "Stabilized Vegas," in *Proc. of 39th Annual Allerton Conference on Communication, Control, and Computing*, October 2002. http://netlab.caltech.edu.

[17] D. H. Choe and S. H. Low, "Stabilized Vegas," in *Proceedings of IEEE Infocom*, April 2003.

[18] F. Paganini, Z. Wang, S. H. Low, and J. C. Doyle, "A new TCP/AQM for stability and performance in fast networks," in *Proc. of 39th Annual Allerton Conference on Communication, Control, and Computing*, October 2002.

[19] Z. Wang and F. Paganini, "Global stability with time delay in network congestion control," in *Proc. of IEEE Conference on Decision and Control*, December 2002.

[20] K. B. Kim and W. H. Kwon, "Intervalwise receding horizon H_∞ tracking controls for linear continuous time-varying systems," in *Proc. of American Control Conference*, (Anchorage, Alaska, USA), pp. 2186–2191, May, 2002.

[21] H. K. Khalil, *Nonlinear systems*. Prentice Hall, 1996.

Table 1: Utilization (%)

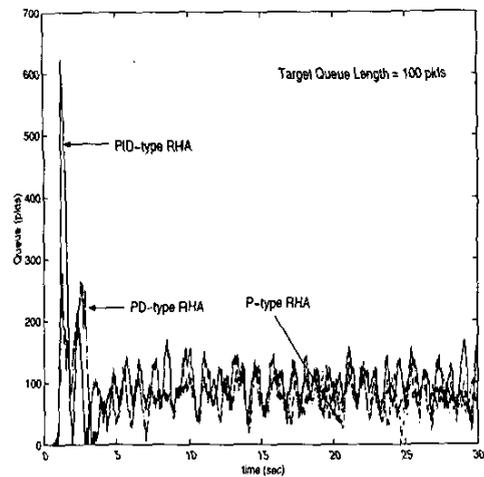| d_s	150msec	250msec	350msec
Memoryless RHA	96.98	93.10	86.21
Memory RHA	96.81	94.83	88.19

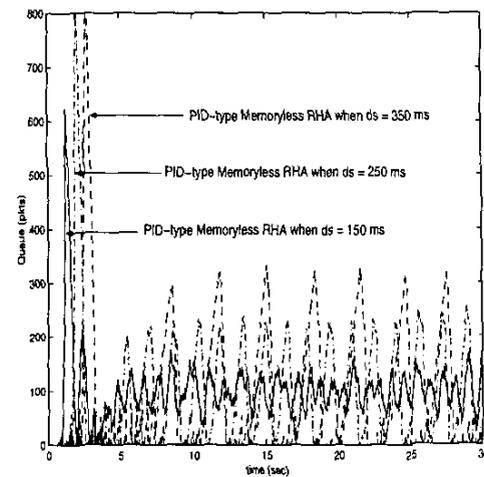**Figure 1:** Queue $b(t)$ trajectory without a memory control
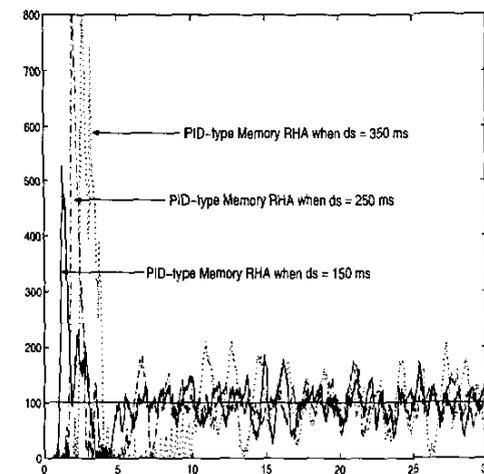

Figure 2: Queue $b(t)$ trajectory without a memory control

Figure 3: Queue $b(t)$ trajectory with a memory control