

Design of AQM in Supporting TCP Based on the Well-Known  
AIMD Model\*

Ki Baek Kim, Ao Tang, Steven H. Low

†Departments of CS and EE  
California Institute of Technology, Pasadena, CA 91125  
kkb@cs,{aotang,slow}@caltech.edu

**Abstract**

In this paper, we investigate how to design AQM with a low-pass filter (average queuing) in supporting TCP based on the well-known AIMD dynamic model. Since we formulate the AQM design problem for the given TCP as state-space models, we get three important features. First, we derive PD-type (Proportional-Derivative) AQM structure with a low-pass filter which includes P-type (Proportional) RED in terms of queue length. Second, we compensate for delays in congestion measure explicitly by adding a memory control structure that uses the previous dynamic information. Third, we obtain a stabilizing optimal gains of the proposed AQM structure by minimizing a linear quadratic cost of the transients on queue length, aggregate rate, and congestion measure. Finally, we illustrate the above theoretical results through *ns* simulations for TCP Reno.

---

\*We submitted this paper to IEEE Globecom, Feb 14, 2003. The corresponding author, Ki Baek Kim will move to France Inria

# 1 Introduction

Congestion control is a distributed iterative procedure to fully utilize the given network resources with fairness by approaching the equilibrium point in a short time. It consists of local algorithms executed dynamically at sources (Transmission Control Protocol, or TCP) and at links (Active Queue Management, or AQM). Since TCP Reno/AQM Droptail has been proposed in [1], the current internet is still using Reno/Droptail and its variants as a congestion control strategy.

Droptail, which drops packets when the queue is full, is not appropriate for high-speed networks since gateways are likely to have large queues in high-speed networks and this would significantly increase the average delay in the network. Furthermore, Droptail can often cause the global synchronization [2, 3] and thus have low throughput. In order to overcome these problems, RED has been suggested in [4] which randomly drops packets proportionally to the average queue length. Since then, there have been a lot of investigations about how to tune design parameters in RED [5, 6, 7, 8]. They show that RED is not enough to stabilize the given TCP and thus not easy to fully utilize the given network resources. As a result, new AQM algorithms such as BLUE [9], AVQ [10] and REM [11] have been suggested. However, all these papers still lack to address how to stabilize the dynamic behavior of TCP and queue systematically in spite that stabilizing the dynamic behavior corresponds to maximizing throughput.

In order to address this problem, the paper [12] has developed a dynamic model to reflect AIMD (Additive Increase Multiplicative Decrease) mode of TCP Reno and the paper [13] has applied the transfer function approach to the RED design problem for stabilizing the AIMD model in [12]. As a sequence, the papers [14, 15] have investigated how to design REM and AVQ for stabilizing the given TCP, respectively. The papers [16, 17] have suggested PI-type (Proportional Integral) AQM in terms of rate in an inner loop. However, all these results based on the transfer function approach do not consider what is a natural state-feedback control [18] in AQM to stabilize the given TCP and queue dynamics. Furthermore, they cannot compensate for delays explicitly since they use only the current dynamic information even if they know the previous dynamic information and delays, where this kind of control strategy is called memoryless control in the control literature. It is well-known in the control literature that the memoryless control has a limit on performance in the presence of a large delay [19, 21], i.e., we cannot fully utilize the given network resources.

Recently, the papers [22, 23] formulate the AQM design problem as state-space models for a single link and homogeneous delays and for multiple links and heterogeneous delays, respectively. Thereby, they show that PD-type (Proportional-Derivative) AQM is a natural state-feedback control structure to stabilize Reno for the first time in the networking literature to our knowledge, where RED is P-type (Proportional) AQM in terms of queue length. However, they do not consider a low-pass filter (average queuing) in RED which makes the congestion signal smooth against sudden change of flow. Note that the paper [13] considers only P-type AQM like RED and the papers [14, 15, 16, 17] do not consider low-pass filter in designing a stabilizing AQM, and all of these results do not compensate for delays explicitly. Since a low-pass filter itself has

a dynamics, it can change dynamic behavior of TCP and queue and thus should be considered in the AQM design problem. This motivates our paper.

In this paper, we suggest how to design AQM with a low-pass filter for stabilizing the AIMD model of [12] in the presence of arbitrary delays. As a first step, we study a simplified version of this problem which can be extended to more realistic problem formulation using modern control theories and technologies, simplified in three regards as in [13, 14, 16, 17, 22]. First, we assume single link, homogeneous delays, and no short flows. Second, we consider the linearized version of the TCP function, so the variables denote perturbations around an equilibrium and the cost measures the deviation from the equilibrium point. For example, a slower transient will incur a higher cost. Third, when we try to get stabilizing optimal gains of AQM, we assume that we know the global information of the given networks accurately. Then, we formulate the AQM design problem as state-space models as in [22, 23]. Thereby, we derive PD-type AQM structure with a low-pass filter, compensate for delays in congestion measure explicitly by using a memory control structure, and obtain a stabilizing optimal AQM by minimizing the linear quadratic cost of the transients on queue length, aggregate rate, and congestion measure. Finally, we verify our results for the nonlinear version of TCP Reno via *ns* simulations.

## 2 Design of a stabilizing AQM with a low-pass filter

In this paper, we use the AIMD model of TCP Reno in [12] for the AQM design problem. Consider the simple case of a single link of capacity  $c$  shared by  $N$  TCP Reno sources and queue dynamics modeled by

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

where  $w_s(t)$  is (expected) TCP window size, in packets, of source  $s$  at time  $t$ ,  $\tau_s^f(t)$  is the forward delay from source to router,  $\tau_s^b(t)$  is the backward delay from router to source,  $\tau_s(t)$  is the round trip time ( $\tau_s(t) = \tau_s^f(t) + \tau_s^b(t)$ ),  $p(t)$  is the loss probability at time  $t$  that is implemented at links,  $b(t)$  is the queue length at time  $t$ ,  $y(t)$  is aggregate rate, and  $c$  is the link capacity, in packets/sec. In the above model, the first and second parts of the first equation represent AI and MD behavior at sources, respectively, where the second equation represents the queue dynamics at link.

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 [13], 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  $\tau_s(t) = \tau_s^b(t)$ . Let  $(w^*, b^*, p^*)$  be the equilibrium point. Then the equilibrium round trip time  $\tau$  is related to  $b^*$  by  $\tau = d + b^*/c$ .

From (1), we can derive the following model of the linearized TCP and queue dynamics:

$$\delta\ddot{b}(t) = A_1\delta b(t) + A_2\delta\dot{b}(t) + B_1\delta p(t - \tau), \quad (2)$$

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

$$\begin{aligned} \delta b(t) &= b(t) - b^*, & \delta\dot{b}(t) &= \dot{b}(t), & \delta\ddot{b}(t) &= \ddot{b}(t), & \delta p(t) &= p(t) - p^* \\ A_1 &= -\frac{2cN}{\tau(2N^2 + c^2\tau^2)}, & A_2 &= -\frac{2cN\tau + 2N^2 + c^2\tau^2}{\tau(2N^2 + c^2\tau^2)}, & B_1 &= -\frac{2N^2 + c^2\tau^2}{2\tau^2 N}. \end{aligned} \quad (3)$$

Refer to [22] for the derivation of (2). Note that the paper [22] does not consider the following averaging queue dynamics in RED.

Consider the following low-pass filter with the average queue length  $\hat{b}(t)$  in RED:

$$\dot{\hat{b}}(t) = -P_1\hat{b}(t) + P_1b(t), \quad P_1 > 0. \quad (4)$$

$P_1$  is a design parameter which decides the cut-off frequency [18]. From (2), we can get

$$\delta\ddot{\hat{b}}(t) = -P_1\delta\ddot{b}(t) + A_1(\delta\dot{\hat{b}}(t) + P_1\delta\hat{b}(t)) + A_2(\delta\ddot{b}(t) + P_1\delta\dot{\hat{b}}(t)) + P_1B_1\delta p(t - \tau), \quad (5)$$

where  $\delta\ddot{\hat{b}}(t) = \ddot{\hat{b}}(t)$ ,  $\delta\ddot{b}(t) = \ddot{b}(t)$ ,  $\delta\dot{\hat{b}}(t) = \dot{\hat{b}}(t)$ , and  $\delta\hat{b}(t) = \hat{b}(t) - b^*$ .

Thus, we can obtain a state-space model for the linearized TCP with a low-pass filter

$$\dot{z}(t) = Az(t) + B\delta p(t - \tau), \quad (6)$$

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

$$z(t) = \begin{bmatrix} \delta\hat{b}(t) \\ \delta\dot{\hat{b}}(t) \\ \delta\ddot{\hat{b}}(t) \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ A_1P_1 & A_1 + A_2P_1 & A_2 - P_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ P_1B_1 \end{bmatrix}. \quad (7)$$

The above state-space model is a minimal representation with state-space variables  $(\delta\hat{b}(t), \delta\dot{\hat{b}}(t), \delta\ddot{\hat{b}}(t))$  of the given differential equation (5). The control strategy based on a state-space model is called modern control in the control literature, while that based on a transfer function model as in [13, 14, 15, 16, 17] is called classical control. Since the state-space approach has been developed in 1950s, it has been widely investigated due to many advantages over the transfer function approach. Please refer to any control literature for more details [18].

From the above state-space model, we can naturally get the following state-feedback AQM

$$\delta p(t) = H z(t) = H_1\delta\hat{b}(t) + H_2\delta\dot{\hat{b}}(t) + H_3\delta\ddot{\hat{b}}(t) \quad (8)$$

if we use only the current dynamic information  $\hat{b}(t)$ ,  $\dot{\hat{b}}(t)$  and  $\ddot{\hat{b}}(t)$  (or  $b(t)$  and  $\dot{b}(t)$ ) at the current time  $t$ . Note that this structure (8) is not captured in the networking literature. How to obtain a pair of stabilizing gain  $(H_1, H_2, H_3)$  is discussed later in this paper.

If we use a memoryless control, i.e., we use only the current state information  $z(t)$  for the delayed control  $\delta p(t - \tau)$ , it is well-known that it is difficult to stabilize a dynamical system in the presence of a large delay [19, 21]. Thus, we cannot fully utilize the given network resources in the presence of large delays  $\tau$ . Next, we overcome this problem by compensating for delays explicitly as far as we know the exact dynamic information.

## 2.1 Explicit delay compensation with a memory control structure

As one way to compensate for large delays in congestion measure explicitly, we need a delay-dependent control that uses not only the current dynamic information but also the previous dynamic information for  $\delta p(t)$  or  $\delta \dot{p}(t)$ . To this end, throughout the rest of this paper, we define

$$\begin{aligned}
\hat{A}_1 &= A_1 + A_2 P_1, & \hat{A}_2 &= A_2 - P_1, & \hat{A}_3 &= A_1 P_1, & \hat{B}_1 &= T_2 \hat{A}_3 \tilde{B}_1 + T_3 \tilde{B}_2 + T_4 \tilde{B}_3 \\
\tilde{B} &= [\tilde{B}_1^T, \tilde{B}_2^T, \tilde{B}_3^T]^T = e^{-A\tau} B \\
T_1 &= -\frac{T_2(\hat{A}_3 \tilde{B}_1 + \hat{A}_1 \tilde{B}_1 \tilde{B}_2^{-1} \tilde{B}_3 + \hat{A}_2 \tilde{B}_3 - \tilde{B}_2^{-1} \tilde{B}_3^2)}{D_1} \quad \text{for some } T_2 \neq 0 \\
T_3 &= T_2 \hat{A}_1 + \frac{T_2(\hat{A}_3 \tilde{B}_2 - \hat{A}_3 \tilde{B}_1 \tilde{B}_2^{-1} \tilde{B}_3)}{D_1}, & T_4 &= T_1 + \hat{A}_2 T_2 \\
D_1 &= \hat{A}_3 \tilde{B}_1^2 \tilde{B}_2^{-1} + \hat{A}_1 \tilde{B}_1 + \hat{A}_2 \tilde{B}_2 - \tilde{B}_3.
\end{aligned} \tag{9}$$

The key to deriving an explicit memory control for the delayed system (6) is to transform the delayed system (6) to the equivalent nominal system

$$\dot{s}(t) = As(t) + \hat{B}\delta p(t), \tag{10}$$

where

$$\begin{aligned}
s(t) &= [s_1, s_2, s_3]^T, \quad \hat{B} = [0, 0, \hat{B}_1]^T \\
s_1(t) &= [T_1 - (T_3 - T_2 \hat{A}_1) \hat{A}_3^{-1} \hat{A}_1](\delta \hat{b}(t) + u_{1\tau}(t)) + [T_2 - (T_3 - T_2 \hat{A}_1) \hat{A}_3^{-1} \hat{A}_2] \\
&\quad (\delta \dot{\hat{b}}(t) + u_{2\tau}(t)) + (T_3 - T_2 \hat{A}_1) \hat{A}_3^{-1} (\delta \ddot{\hat{b}}(t) + u_{3\tau}(t))
\end{aligned} \tag{11}$$

$$s_2(t) = (T_3 - T_2 \hat{A}_1)(\delta \hat{b}(t) + u_{1\tau}(t)) + T_1(\delta \dot{\hat{b}}(t) + u_{2\tau}(t)) + T_2(\delta \ddot{\hat{b}}(t) + u_{3\tau}(t)) \tag{12}$$

$$s_3(t) = T_2 \hat{A}_3(\delta \hat{b}(t) + u_{1\tau}(t)) + T_3(\delta \dot{\hat{b}}(t) + u_{2\tau}(t)) + T_4(\delta \ddot{\hat{b}}(t) + u_{3\tau}(t)) \tag{13}$$

$$\begin{bmatrix} u_{1\tau}(t) \\ u_{2\tau}(t) \\ u_{3\tau}(t) \end{bmatrix} = \int_{-\tau}^0 e^{-A(\sigma+\tau)} B \delta p(t+\sigma) d\sigma. \tag{14}$$

Refer to Appendix A for the derivation of (10). Note that the nominal system is asymptotically stable if and only if the original delayed system (6) is asymptotically stable.

From the above state-space model, we can get a memory state-feedback AQM

$$\delta p(t) = H_1^T(\delta \hat{b}(t) + u_{1\tau}(t)) + H_2^T(\delta \dot{\hat{b}}(t) + u_{2\tau}(t)) + H_3^T(\delta \ddot{\hat{b}}(t) + u_{3\tau}(t)). \tag{15}$$

Next, we show how to obtain optimal gains of the proposed AQM structure.

## 2.2 Stabilizing Optimal AQM

As a performance measure for (10), we consider the following optimization problem:

$$\min_{\delta p(\cdot)} J(s(t), \delta p(\cdot)) = \int_t^{t+\infty} (s^T(\sigma) Q s(\sigma) + \delta p^2(\sigma)) d\sigma, \tag{16}$$

where  $Q = Q^T \geq 0$ .

For ease of explanation of the above performance index, assume that  $d = 0$  in (16). Then, we define the optimal AQM design as the problem of choosing an input  $\delta p(t)$  that minimizes the cost of transient around an equilibrium:

$$\min_{\delta p(\cdot)} J(z(t), \delta p(\cdot)) = \int_t^{t+\infty} \left( Q_1 \delta \hat{b}^2(\sigma) + Q_2 \delta \dot{\hat{b}}^2(\sigma) + Q_3 \delta \ddot{\hat{b}}^2(\sigma) + \delta p^2(\sigma) \right) d\sigma.$$

Each term in the integrand penalizes average-queue length, average-queue length rate, and the fluctuation of the loss probability, respectively. Hence the cost is a weighted sum of transients on average queue, average-queue rate, and fluctuation in probability, weighted by  $Q_1 > 0$ ,  $Q_2 > 0$ , and  $Q_3 > 0$ .

**Proposition 1** *If we solve the problem (16), then we get a stabilizing optimal AQM*

$$\delta p^*(t) = H_1^* s_1(t) + H_2^* s_2(t) + H_3^* s_3(t), \quad (17)$$

where  $H_1^* > -\frac{\hat{A}_3}{\hat{B}_1}$ ,  $H_2^* > -\frac{\hat{A}_1}{\hat{B}_1}$ ,  $H_3^* > 0$ ,

$$H_1^* = -\frac{\hat{A}_3 + \sqrt{\hat{A}_3^2 + \hat{B}_1^2 Q_1}}{\hat{B}_1}, \quad H_2^* = \frac{-2\hat{A}_2 H_3^* - \hat{B}_1 H_3^{*2} + \hat{B}_1 Q_3}{2},$$

and  $H_3^*$  is the positive solution of the following fourth order polynomial:

$$\begin{aligned} & -\hat{B}_1^4 H_3^{*4} - 4\hat{A}_2 \hat{B}_1^3 H_3^{*3} + (4\hat{A}_1 \hat{B}_1^2 - 4\hat{A}_2^2 \hat{B}_1^2 + 2\hat{B}_1^4 Q_3) H_3^{*2} - [8\hat{B}_1(\hat{A}_3 + \sqrt{\hat{A}_3^2 + \hat{B}_1^2 Q_1}) - 8\hat{A}_3 \hat{B}_1 \\ & - 8\hat{A}_1 \hat{A}_2 \hat{B}_1 - 4\hat{A}_2 \hat{B}_1^3 Q_3] H_3^* - 8\hat{A}_2(\hat{A}_3 + \sqrt{\hat{A}_3^2 + \hat{B}_1^2 Q_1}) - 4\hat{A}_1 \hat{B}_1^2 Q_3 + 4\hat{B}_1^2 Q_2 - \hat{B}_1^4 Q_3^2 = 0. \end{aligned}$$

**Proof:** The optimal control that minimizes (16) is given by  $u^*(t) = H^* s(t)$  [24].  $\square$

Proposition 1 implies that the solution of the problem (16) is an AQM algorithm, specified by  $(H_1^*, H_2^*, H_3^*)$ . Conversely, given any AQM of this structure, it solves the problem (16) with appropriate weights  $Q_i$ , as the next result says. It can be easily proved from Proposition 1.

**Proposition 2** *Given a stabilizing AQM  $\delta p(t) = [H_1 \ H_2 \ H_3]s(t)$ , it solves the problem (16) with weights*

$$\begin{aligned} Q_1 &= \frac{H_1(2\hat{A}_3 + \hat{B}_1 H_1)}{\hat{B}_1}, \quad Q_2 = H_2^2 + 2 \frac{\hat{A}_1 \hat{B}_1 H_2 + (\hat{A}_2 + \hat{B}_1 H_3) \sqrt{\hat{A}_3^2 + \hat{B}_1^2 Q_1} + \hat{A}_2 \hat{A}_3}{\hat{B}_1^2} \\ Q_3 &= H_3^2 + 2 \frac{\hat{A}_2 H_3 + H_2}{\hat{B}_1}. \end{aligned}$$

**Proposition 3** *Given the eigenvalues  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  of the closed-loop system (10) with (17), where real parts of  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are negative,  $\delta p(t)$  solves the problem (16) with weights*

$$\begin{aligned} Q_1 &= \frac{\hat{\lambda}_3^2 - \hat{A}_3^2}{\hat{B}_1^2}, \quad Q_2 = \frac{-\hat{A}_1^2 + \hat{\lambda}_2^2 + 2\hat{\lambda}_1 \sqrt{\hat{A}_3^2 + \hat{B}_1^2 Q_1} + 2\hat{A}_2 \hat{A}_3}{\hat{B}_1^2} \\ Q_3 &= \frac{-\hat{A}_2^2 - 2\hat{A}_1 + \hat{\lambda}_1^2 - 2\hat{\lambda}_2}{\hat{B}_1^2}, \end{aligned}$$

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

We now interpret RED as an approximation of the proposed AQM. For ease of comparison, we assume that  $\tau = 0$  (i.e.,  $s(t) = z(t)$ ,  $\hat{B} = B$ ) for linearized model.

**Remark 1** *The linear model motivated by RED is:*

$$RED: \quad \delta p^r(t) = H_1^r \delta \hat{b}(t)$$

for some nonnegative constants  $H_1^r$ . By Proposition 1, the proposed AQM has  $\delta p(t) = H_1 \delta \hat{b}(t) + H_2 \delta \hat{\dot{b}}(t) + H_3 \delta \hat{\ddot{b}}(t)$ . Thus, if we set  $H_1 = H_1^r$ ,  $H_2 = 0$  and  $H_3 = 0$ , then we can get the above RED. Note that  $H_3$  should be positive by Proposition 1. From Propositions 1-3, the sum of eigenvalues of the closed-loop system with RED is given by

$$\hat{\lambda}_1 = \hat{A}_2 + \hat{B}_1 H_3 \leq \hat{A}_2,$$

where the last inequality follows from that  $\hat{A}_2 < 0$ ,  $\hat{B}_1 < 0$ , and  $H_3 > 0$ . 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  $H_3 = 0$  than when  $H_3 > 0$ . This suggests that the decay rate is smaller with RED ( $H_2 = H_3 = 0$ ).

### 3 ns simulations for TCP Reno

In this section, we illustrate performance of the derived structures via two simulation examples. In each example, we compare the proposed PD-type memoryless AQM with RED and compare the proposed memory PD-type AQM with the proposed memoryless one in the presence of large propagation delays.

For ease of implementation of stabilizing optimal AQMs, we set all eigenvalues of the closed-loop system to be equal (i.e.,  $\lambda = \lambda_i$  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 2.2. RED-like AQM is obtained by setting  $H_2 = H_3 = 0$  after getting gains  $H_1$ ,  $H_2$ , and  $H_3$  of the proposed stabilizing optimal AQM.

Similarly to [13, 14], we simulate a single bottleneck link with capacity  $c = 4000$  pkts/sec shared by  $N = 100$  Reno sources. The AQM at the bottleneck link uses ECN marking. Real queue length is limited by  $b_{\max} = 800$  pkts. Marking probability of each AQM is updated every 2ms, i.e., the sampling time is  $T_s = 2$ ms, where  $\hat{b}(t) = \hat{b}(t - T_s) + T_s P_1(-\hat{b}(t - T_s) + b(t - T_s))$ . The simulation duration is 40 sec.

#### 3.1 RED-like AQM and the proposed memoryless PD-type AQM

In this example, we compare performances of RED-like AQM and the proposed memoryless PD-type AQM without explicit delay compensation in the presence of delays. We assume that

the propagation delay is  $d_s = 150\text{ms}$ , where the round trip time is  $\tau^* = d_s + b^*/c$  and the target queue length is  $b^* = 200$  pkts.

The memoryless AQM of (17), which is obtained by setting  $\tau = 0$ , is implemented by

$$p(t) = p^* + H_1(\hat{b}(t) - b^*) + H_2 P_1(-\hat{b}(t) + b(t)) + H_3 P_1[P_1(\hat{b}(t) - b(t)) + y(t) - y(t_0)],$$

where  $H_1 = -\frac{(\hat{A}_3 - \lambda^3)}{B_1}$ ,  $H_2 = -\frac{(\hat{A}_1 + 3\lambda^2)}{B_1}$ , and  $H_3 = -\frac{(\hat{A}_2 - 3\lambda)}{B_1}$ .

Figure 1 shows their performances when  $\lambda = -3.75$ . It illustrates Remark 1 that the proposed AQM is easier to stabilize TCP than RED-like AQM. The average queue lengths of RED-like AQM and the proposed AQM are 434.8 pkts and 205.3 pkts, respectively and utilization of RED-like AQM and the proposed AQM are 93 % and 95 %, respectively. Thus, the proposed PD-type AQM has a smaller queuing delay but a larger throughput than RED-like AQM.

### 3.2 Performance of a memory stabilizing AQM

In this example, we illustrate the performance of a memory control structure by comparing a memory PD-type AQM with a memoryless one. Here, we hold all simulation parameters of the previous subsection except  $d_s = 350\text{msec}$ .

The proposed memory AQM (17) is implemented by

$$\begin{aligned} p(t) = & p^* + H_1^\tau(\hat{b}(t) - b^* + u_{1\tau}(t)) + H_2^\tau(-P_1\hat{b}(t) + P_1b(t) + u_{2\tau}(t)) \\ & + H_3^\tau[P_1^2\hat{b}(t) - P_1^2b(t) + P_1(y(t) - y(t_0)) + u_{3\tau}(t)], \end{aligned}$$

where  $H_1^\tau = H_1^*[T_1 - (T_3 - T_2\hat{A}_1)\hat{A}_3^{-1}\hat{A}_1] + H_2^*(T_3 - T_2\hat{A}_1) + H_3^*T_2\hat{A}_3$ ,  $H_2^\tau = H_1^*[T_2 - (T_3 - T_2\hat{A}_1)\hat{A}_3^{-1}\hat{A}_2] + H_2^*T_1 + H_3^*T_3$ ,  $H_3^\tau = H_1^*(T_3 - T_2\hat{A}_1)\hat{A}_3^{-1} + H_2^*T_2 + H_3^*T_4$ ,  $H_1^* = -\frac{(\hat{A}_3 - \lambda^3)}{B_1}$ ,  $H_2^* = -\frac{(\hat{A}_1 + 3\lambda^2)}{B_1}$ , and  $H_3^* = -\frac{(\hat{A}_2 - 3\lambda)}{B_1}$ .

Figure 2 illustrates that the proposed memory AQM is much closer to the target queue length ( $b^* = 200$  pkts) than the proposed memoryless AQM as the propagation delay increases, while they have similar performances when the delay is small. Utilization of the memoryless AQM and the memory AQM are 83 % and 91 %, respectively. Thus, the proposed memory AQM has a smaller queuing delay but a larger throughput than the proposed memoryless AQM. We have the same results for other large propagation delays.

## 4 Conclusion

In this paper, we investigated the design of AQM with a low-pass filter (average queuing) for stabilizing the given TCP. Thereby, we derived PD-type AQM, compensated for delays in congestion measure explicitly by adding a memory control structure, and obtained stabilizing optimal gains of the proposed AQM structure. Finally, we illustrated the proposed results via *ns* simulations. In simulations, we used only one parameter ( $\lambda$ ) for designing the proposed AQMs.

Our result based on the AIMD model in [12] shows that P-type AQM like RED is not enough to stabilize the AIMD dynamic behavior of TCP Reno. We expect that this work will be helpful



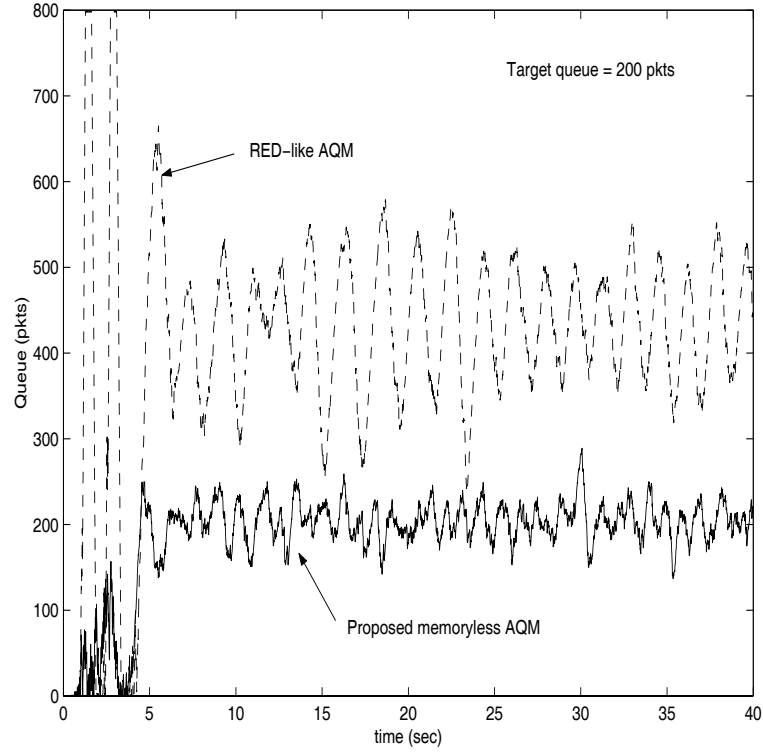


Figure 1: Queue  $b(t)$  trajectories of P- and PD-type memoryless AQMs

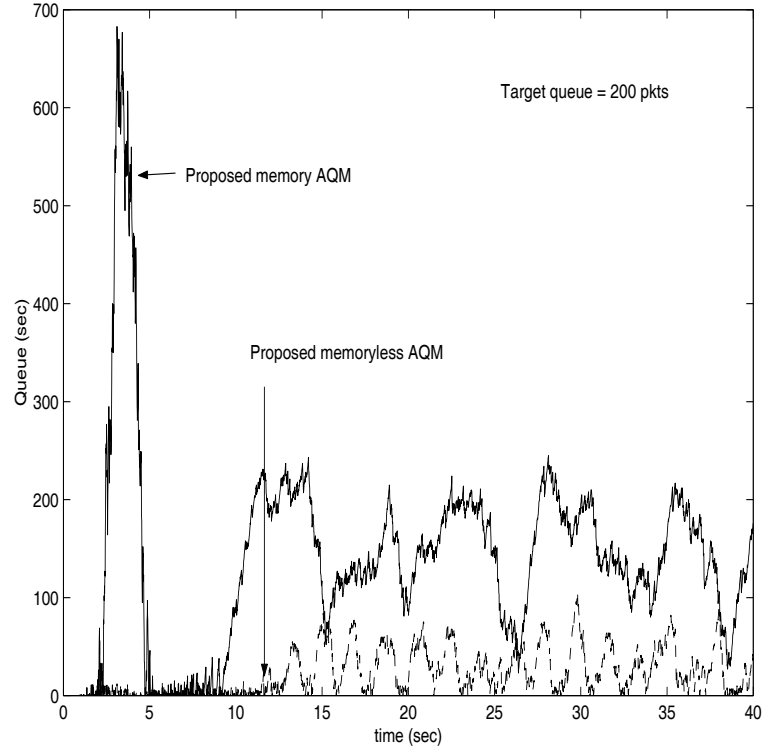


Figure 2: Queue  $b(t)$  trajectories of the memoryless and memory AQMs

to design a new AQM control structure which fully supports any kind of TCP in the future. For this purpose, we can think about four directions of future research. First, we need to extend the proposed results to more realistic models. Second, it is necessary to consider how to estimate the number of sources and round-trip times at routers and how to implement the memory control structure in real-time. Third, we should investigate how to design a robust AQM in the presence of uncertainty in estimation of system parameters and delays. Finally, it would be interesting to approximate the stabilizing optimal AQM in a decentralized manner.

## References

- [1] V. Jacobson, "Congestion avoidance and control," in *Proceedings of ACM/SIGCOMM*, 1988. <ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z>.
- [2] F. Baccelli and D. Hong, "AIMD, fairness and fractal scaling of TCP traffic," tech. rep., INRIA, Paris, France, 2001. RR 4155.
- [3] F. Baccelli and D. Hong, "Interaction of tcp flows as billiards," tech. rep., INRIA, Paris, France, 2002. RR 4437.
- [4] 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>.
- [5] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A self-configuring RED gateway," in *Proceedings of IEEE/INFOCOM*, March 1999. <http://www.eecs.umich.edu/~wuchang/work/infocom99.ps.Z>.
- [6] T. J. Ott, T. V. Lakshman, and L. Wong, "SRED: Stabilized RED," in *Proceedings of IEEE Infocom'99*, March 1999. <ftp://ftp.bellcore.com/pub/tjo/SRED.ps>.
- [7] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith, "Tuning RED for web traffic," in *Proceedings of ACM/SIGCOMM*, 2000.
- [8] M. May, T. Bonald, and J.-C. Bolot, "Analytic evaluation of RED performance," in *Proceedings of IEEE Infocom*, March 2000.
- [9] W. Feng, K. G. Shin, D. Kandlur, and D. Saha, "Stochastic Fair Blue: A queue management algorithm for enforcing fairness," in *Proceedings of INFOCOM*, April 2001.
- [10] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: utility functions, random losses and ECN marks," in *Proceedings of IEEE/INFOCOM*, 2000. <http://www.ieee-infocom.org/2000/papers/401.ps>.
- [11] 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>.

- [12] 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 *Proceedings of ACM SIGCOMM*, September 2000.
- [13] 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>.
- [14] 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>.
- [15] 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>.
- [16] H. Lim, K. J. Park, E. C. Park, and C. H. Choi, "Active queue management algorithm with a rate regulator," in *Proc. of 15th IFAC World Congress on Automatic Control*, vol. T-We-M02, (Barcelona, Spain), 2002.
- [17] E. C. Park, H. Lim, K. J. Park, and C. H. Choi, "Anaysis of the virtual rate control algorithm in tcp networks," in *Globecom*, vol. Nov, (Taipei, Taiwan), 2002.
- [18] G. F. Franklin, J. D. Powell, and A. E. Naeini, *Feedback Control of Dynamic Systems*. Addison-Wesley, 1994.
- [19] J. H. Lee, S. W. Kim, and W. H. Kwon, "Memoryless  $H_\infty$  controllers for state delayed systems," *IEEE Trans. on Autom. Contr.*, vol. 39, pp. 159 – 162, 1994.
- [20] L. Xi and C. E. de Souza, "Delay-dependent robust stability and stabilization of uncertain linear delay systems," *IEEE Trans. Automat. Contr.*, vol. 42, no. 8, pp. 1144–1148, 1997.
- [21] P. Park, "A delay-dependent stability criterion for systems with uncertain time-invariant delays," 1999.
- [22] K. B. Kim and S. H. Low, "Analysis and design of AQM for stabilizing TCP." To appear in American Control Conference'03, Caltech Technical Report caltechCSTR:2002.009, March 2002. <http://cisl.snu.ac.kr/~kbb>, <http://caltechcstr.library.caltech.edu/>, <http://netlab.caltech.edu>.
- [23] 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/~kbb>, <http://netlab.caltech.edu>.
- [24] D. E. Kirk, *Optimal Control Theory*. Englewood Cliffs, New Jersey: Prentice-Hall, 1970.

## A Derivation of (10)

We note that the system (6) can be written as  $z(t + \tau) = e^{A\tau}[z(t) + \int_{-\tau}^0 e^{-A(\sigma+\tau)} B \delta p(\sigma + t) d\sigma]$ . Define  $h(t) = z(t) + \int_{-\tau}^0 e^{-A(\sigma+\tau)} B \delta p(t + \sigma) d\sigma = z(t) + \begin{bmatrix} u_{1\tau}(t) \\ u_{2\tau}(t) \\ u_{3\tau}(t) \end{bmatrix}$ . Then, the system (6) can be rewritten as

$$\dot{h}(t) = Ah(t) + \tilde{B} \delta p(t), \quad (\text{A.18})$$

where  $\tilde{B} = [\tilde{B}_1^T, \tilde{B}_2^T, \tilde{B}_3^T]^T = e^{-A\tau} B$ .

Let  $s(t) = Th(t)$ , where  $T = \begin{bmatrix} T_1 - (T_3 - T_2 \hat{A}_1) \hat{A}_3^{-1} \hat{A}_1 & T_2 - (T_3 - T_2 \hat{A}_1) \hat{A}_3^{-1} \hat{A}_2 & (T_3 - T_2 \hat{A}_1) \hat{A}_3^{-1} \\ T_3 - T_2 \hat{A}_1 & T_1 & T_2 \\ T_2 \hat{A}_3 & T_3 & T_4 \end{bmatrix}$ .  $T_2$  should be selected so that  $\det(T) \neq 0$  ( $T_2$  cannot be zero). Then, the system (A.18) can be rewritten as  $\dot{s}(t) = TAT^{-1}s(t) + T\tilde{B}\delta p(t) = As(t) + \begin{bmatrix} 0 \\ 0 \\ \hat{B}_1 \end{bmatrix} \delta p(t)$ , where  $\hat{B}_1 = T_2 \hat{A}_3 \tilde{B}_1 + T_3 \tilde{B}_2 + T_4 \tilde{B}_3$ .