INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

H2-AQM - An Optimal Active Queue Management Controller

Michele M. de A. E. Lima
Nelson L. S. da Fonseca *José C. Geromel*

Technical Report - IC-03-009 - Relatório Técnico

April - 2003 - Abril

H2-AQM - An Optimal Active Queue Management Controller

Michele M. de A. E. Lima* † Nelson L. S. da Fonseca† José C. Geromel‡

Abstract

This report introduces a novel AQM policy that uses an H_2 optimal controller. The synthesis of the controller uses a non-rational approach, in which the stability and performance objectives of the system are completely expressed as Linear Matrix Inequalities (LMIs). The controller stabilizes the system under diverse network conditions.

1 Introduction

The Transmission Control Protocol (TCP) changes its transmission rate according to the estimated available bandwidth. Such change is governed by the receipt of acknowledgements from the receiver. If three acknowledgments for the same packet are received, the packet is considered lost and the transmission window is reduced by half. If an acknowledgment is not received after a certain period (timeout), the packet is also considered lost, and the transmission window is drastically reduced to one packet. During the elapsed time between the loss of a packet and the detection of the loss by the sender, numerous packet might have been transmitted and dropped, increasing congestion and wasting bandwidth. To ameliorate such problem, Active Queue Management (AQM) mechanisms have been proposed.

The idea behind AQM is the early notification of incipient congestion so that TCP senders can reduce their transmission rate before queue overflows, avoiding the degradation of TCP performance.

Random Early Detection policy (RED) [1] is the AQM policy recommended by the Internet Engineering Task Force (IETF) to be deployed in the Internet [2]. RED was proposed to avoid congestion, to ensure an upper bound on the average queue size, to avert global synchronization and to prevent bias against bursty traffic. RED estimates the average queue size, which is compared to two threshold values: min_{th} and max_{th}. If the average queue size is smaller than min_{th}, no packet is marked/dropped. If it is between min_{th} and max_{th} each arriving packet is marked/dropped with probability p_a, which grows linearly with the estimated queue size. Otherwise, every arriving packet is dropped. Setting RED parameters is a major challenge. When threshold values are not correctly defined, RED can perform even worst than the traditional tail drop policy. When min_{th} is too low, the link may be underutilized and when max_{th} is too high delay increases unnecessarily.

*Department of Informatics - State University of West Paraná - (UNIOESTE)
†Institute of Computing - State University of Campinas - (UNICAMP)
‡Faculty of Electrical and Computer Engineering - State University of Campinas - (UNICAMP)

The performance of TCP over RED is sensitive to the number of active connections. When there is a great number of active flows, the aggregate traffic is bursty, causing large oscillations in queue size and, consequently, increasing jitter and link underutilization. Moreover, when the mark/drop probability value varies too much in a short period, RED fails to mark the packets randomly, causing the global synchronization phenomenon, which is the reduction of the transmission window size of many connections at the same time in response to the drop of a packet. When the average queue size is larger than max_{th}, all packets are dropped/marked causing significant decrease in throughput.

To overcome the difficulties of tuning RED parameters, numerous studies based on heuristics and simulations have been conducted. Nonetheless, such studies neither assure that an equilibrium point can be reached nor guarantee stability of the queue size [3, 4]. On the other hand, investigations have been carried out to derive RED configuration in a more systematic way [5, 6].

To design and develop AQM policies that ensure stability about an equilibrium point some approaches based on Optimization and Control Theory have been proposed. In the former, the purpose is to characterize the equilibrium conditions that can be obtained given a network state. In policies based on Control Theory, congestion control is viewed as a feedback control system. The aim is, thus, to capture the dynamics of congestion and ensure stability of the queue size. In [7], the behavior of AQM queue fed by TCP sources was modelled and in [8] the authors proposed a Proportional Integrator controller and compared its performance to RED.

This paper presents a new AQM policy, that uses an optimal controller. A simplified non-linear dynamic model of TCP presented in [9] is used to design this controller. The model is linearized and the resulting system is linear with constant delay. The output that gives the expected performance is identified, and a controller that produces the optimal solution to achieve the target performance is developed.

The novelty of this paper is the new approach of using non-rational controllers. The use of non-rational controllers overcomes the main difficulty of designing rational controller for linear delay systems, which is to incorporate in the design problem the matrix multiplier used to prove stability with respect to the delayed part of the system [10]. Furthermore, stability and performance objectives are completely expressed as Linear Matrix Inequalities (LMIs).

This paper is organized as follows. Section 2 describes related works. Section 3 shows a TCP model used in the design of the proposed AQM system. In Section 4, the design of H2-AQM, an active queue management algorithm that uses a non-rational controller, is introduced. In Section 5, numerical results are presented. Finally, in Section 6, conclusions are drawn.

2 Related Works

AQM policies based on Optimization Theory aim at maximizing the transmission rates of sources while providing equal share of the available bandwidth among them [11, 12]. Utility functions of the aggregate flows are defined to identify the equilibrium conditions for a network state. In the dual version of this problem, the loss rate is the dual variable. The purpose is to achieve the minimum loss rate that assures the maximum transmission rates. The policy Random Exponential Marking (REM) is presented as a solution for this problem [13]. REM expresses the congestion measures as prices, computed for each link using local information. Price values are informed to the sources through packet dropping/marking.

Policies based on Control Theory consider the intrinsic feedback nature of congestion system. The transmission rate of the sources are adjusted according to the level of congestion, determined by the queue occupancy. The notification of congestion to the sources is done through packet dropping/marking. Therefore, controllers are responsible to determine the appropriate value of the drop/mark probability value that stabilizes the queue size independently of network conditions, which is of paramount importance to avoid jitter and low link utilization.

Stabilized RED (SRED) aim at stabilizing the queue size about a reference value independently of the number of active flows. It estimates the number of active TCP connections and compute the suitable drop/mark probability for that number of flows [14]. Dynamic RED (DRED) has the same objective of SRED. However, it does not estimate the number of active flows. It uses a proportional controller, with a gain determined empirically by simulation [15].

In [8] a simplified model of the dynamic of TCP behavior [7] is used to derive the Proportional Integrator AQM (PI-AQM), which uses a proportional integrator controller. The plant os this controller, however, does not represent all the dynamic of the system, making necessary the establishment of the network conditions for which the controller is able to stabilize the system.

The H2-AQM controller, presented in this work, utilizes the same simplified model used to derive the PI-AQM controller. Nevertheless, the plant used in the H2-AQM design represents the system in greater detail, ensuring system stability independently of the network conditions. Another difference is that H2-AQM uses optimal controllers instead of classical controllers such as proportional or proportional integrator. Moreover, in this work the stability and performance objectives of the AQM system are expressed and solved as Linear Matrix Inequalities (LMIs). In other words, the computation of the controller parameters can be performed by solving a convex problem.

3 A Dynamic Model for the TCP Behavior

A fluid-flow stochastic differential equation model for the TCP behavior was introduced in [7]. A simplified version of this model, which suppress the timeout mechanism of TCP is given by [8]:

$$\begin{aligned}
\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)}{2}\frac{W(t-R(t))}{R(t-R(t))}p(t-R(t)); && (1)\\
\dot{q}(t) &= -C(t) + \frac{N(t)}{R(t)}W(t) + \omega_q(t); && (2)
\end{aligned}$$

$$R(t) = \frac{q(t)}{C(t)} + T_p; \tag{3}$$

$W(t)$: is the average TCP window size in packets;
$q(t)$: is the queue size in packets;
$\omega_q(t)$: is the noise produced by UDP flows;
$R(t)$: is the round trip time (RTT) in seconds;
$C(t)$: is the link capacity in packets/second;
T_p: is the propagation delay in seconds;
$N(t)$: is the load factor in number of TCP connections;
$p(t)$: is the packet mark/drop probability;

Equation (1) describes the TCP window dynamics. The first term models the window additive increase, while the second models its multiplicative decrease. Equation (2), captures the queue behavior as the difference between the arrival rate, $NW/R + \omega_q(t)$, and the link capacity, C.

Equation (2) differs from the one presented in [8] by the term $\omega_q(t)$, which was included to take into account the contribution of UDP flows to the queue size. UDP flows are non-adaptive, which means that they do not reduce their transmission rate under congestion.

Let the number of TCP connections and the capacity be constant, i. e. $N(t) \equiv N$ and $C(t) \equiv C$ and let (W, q) be the system state, p the input, and $R_0 = \frac{q_0}{C} + T_p$, the equilibrium point (W_0, q_0, p_0) is computed by solving $\dot{W}(t) = 0$ and $\dot{q}(t) = 0$, which gives:

$$W_0 = \sqrt{\frac{2}{p_0}} = \frac{R_0 C}{N} = \frac{q_0 + CT_p}{N}; \tag{4}$$

$$q_0 = N\sqrt{\frac{2}{p_0}} - CT_p = CR_0 - CT_p; \tag{5}$$

$$p_0 = \frac{2N^2}{(R_0 C)^2} = \frac{2N^2}{(q_0 + CT_p)^2}; \tag{6}$$

Equations (1) and (2) are linearized about the equilibrium point (W_0, q_0, p_0) resulting in:

$$\begin{aligned} \dot{x}_1(t) &= -\frac{N}{R_0^2 C}(x_1(t) + x_1(t - R_0)) - \frac{R_0 C^2}{2N^2}u(t - R_0) - \frac{1}{R_0^2 C}(x_2(t) - x_2(t - R_0)) \\ \dot{x}_2(t) &= \frac{N}{R_0}x_1(t) - \frac{1}{R_0}x_2(t) \end{aligned} \tag{7}$$

where:

$$\begin{aligned} x_1(t) &\doteq W(t) - W_0; \\ x_2(t) &\doteq q(t) - q_0; \\ u(t) &\doteq p(t) - p_0; \end{aligned}$$

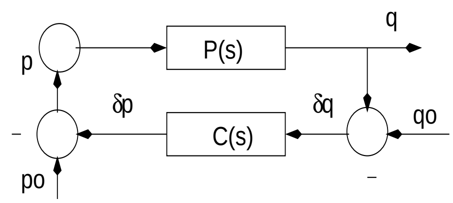

Figure 1: AQM as a feedback control system

4 Design of an Optimal Controller for Active Queue Management

In this section, the congestion control system (7) is represented as a continuous time linear delay system in state space form and a non-rational approach is used to derive the H_2 optimal controller. The synthesis of the controller is based on the results presented in [10], where the design of the controller for linear delay systems is cast and solved as Linear Matrix Inequalities (LMIs).

The TCP dynamics of system (7) can be analyzed as a function of network parameters such as the number of TCP flows, N, the round trip time (RTT), R_0, the link capacity, C, and in terms of the intrinsic feedback nature of AQM [8]. The action of the AQM controller, $C(s)$, is to mark/drop packets with probability value p, using the measured queue size q. It should also stabilize the plant of the system, denoted by the transfer function $P(s)$, which is irrational in s and relates how the mark/drop probability affects the queue size. Figure 1 presents an AQM feedback control system.

The plant $P(s)$ captures the dynamic of the system in greater detail than the one in [8], where the plant is simplified to isolate the contributions due to delay in the residual, $\Delta(s)$. Such residual is treated as an unmodeled dynamics of the system.

The advantage of representing in plant $P(s)$ the complete dynamic of the system is the fact that the stabilization of the plant by the controller $C(s)$ implies in system stabilization.

In [8], to ensure stability of the system the controller $C(s)$ has to stabilize the residual $\Delta(s)$, and also need to establish an upper bound for $\Delta(s)V(s)$, where $V(s)$ is the sensitivity function of the system. It implies in identifying N, R_0 and C values, for which the controller can stabilize the system.

The linear system presented in (7) can be expressed in state space form by the following equations that describe a continuous time linear delay system:

$$\begin{aligned}
\dot{x}(t) &= A_0x(t) + A_1x(t-R_0) + B_ww(t) + B_uu(t-R_0); \\
z(t) &= C_zx(t) + D_{zu}u(t); \\
y(t) &= C_yx(t-R_0) + D_{yw}w(t);
\end{aligned} \tag{8}$$

where $x(t)$ is the state vector; $u(t)$ is the control input that represents the probability $p(t)$; $w(t)$ is the external noise produced by UDP load; $z(t)$ is the reference output, i. e., the desired output for the system and $y(t)$ is the measured output.

Now, consider that the system described in (8) is connected to the following controller:

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{A}_0\hat{x}(t) + \hat{A}_1\hat{x}(t-R_0) + \hat{B}y(t); \\ u(t) &= \hat{C}_0\hat{x}(t) + \hat{C}_1\hat{x}(t-R_0) + \hat{D}y(t); \end{aligned} \tag{9}$$

This controller can also be described in the frequency domain by the non-rational transfer function:

$$C(s) = (\hat{C}_0 + \hat{C}_1 e^{-sR0})(sI - \hat{A}_0 - \hat{A}_1 e^{-sr_0})^{-1}\hat{B} + \hat{D}; \tag{10}$$

The controller (9) was carefully chosen to reproduce the structure of the plant of the system (8). The goal is to determine the matrices of the controller (9) that stabilizes (8) while minimizing a certain measure of the reference output $z(t)$. It is, thus, necessary to define the desired performance goals for the output $z(t)$ and what should be measured in the output $y(t)$. This implies that the desired performance goals for the designed AQM policy must be represented in the controller $C(s)$.

The performance goals of an AQM policy as suggested in [8] are: congestion avoidance, efficient queue utilization and assurance of low delay and delay variation. Using the queue efficiently means that unnecessary periods of overflow and emptiness should be avoided. The former results in loss of packets, undesired retransmissions, and penalization of bursty traffic, while the later implies in buffer underutilization. In order to produce low delay values, the queue size should be small, which, however, can lead to link underutilization. Additionally, queue size variations should be avoid to prevent jitter, which is detrimental to some real time applications. Thus, the ideal mark/drop probability value should assure maximum transmission rates while minimizing the queue size subject to the network conditions, so that needless packet loss is avoided. To achieve such goals, the matrices of system (8) are defined as:

$$A_0 = \begin{bmatrix} -\frac{N}{R_0^2C} & -\frac{1}{R_0^2C} \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}, \quad B_w = \begin{bmatrix} 0 & 0 \\ 0.2C & 0 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} -\frac{N}{R_0^2C} & \frac{1}{R_0^2C} \\ 0 & 0 \end{bmatrix}, \quad B_u = \begin{bmatrix} -\frac{R_0C^2}{2N^2} \\ 0 \end{bmatrix},$$

$$C_z = \begin{bmatrix} 0 & 1 \\ \frac{N}{R_0} & -\frac{1}{R_0} \\ 0 & 0 \end{bmatrix}, \quad D_{zu} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix};$$

$$C_y = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad D_{yw} = \begin{bmatrix} 0 & 0.02C \end{bmatrix};$$

Matrices A_0, A_1 and B_u are obtained directly from the linearization of the system. A_0 represents the terms of the system without delay, while matrix A_1 represents the delay term. The first row of these matrices corresponds to $\frac{\partial f_1}{\partial x_i}$, and the second corresponds to $\frac{\partial f_2}{\partial x_i}$, with $i = 1, 2$. B_u contains $\frac{\partial f_1}{\partial u}$ in the first row, and $\frac{\partial f_2}{\partial u}$ in the second row, where f_1 is the right-hand of Equation (1) and f_2 the right-hand of Equation (2), $x_1 = W$, $x_2 = q$ and $u = p$.

B_w, controls the amount of noise existing in the system, which is generated by UDP flows. The value chosen allows UDP flows to utilize up to 20% of the link capacity. Such value is a satisfactory tolerance margin, given that 95% of the bytes transmitted in the Internet are generated by TCP [16].

C_z translates the design goal, which is to maximize the transmission rates while minimizing the queue size and its variation. The first row is related to the queue size, and the second to the queue variation. D_{zu}, weighs the value of the drop/mark probability in the output. Different weight values were tested, from 0.3 to 0.9. Results were quite similar, so, a value of 0.5 was adopted.

C_y indicates that the value of interest measured in the output is the queue size in the previous RTT. Finally, D_{yw}, weighs the noise in the measured output, which is, in general, 10% of the value in matrix B_w.

Let $\bar{x}(t)$ be the augmented state vector which contains the state vector $x(t)$ and the controller state vector $\hat{x}(t)$:

$$\bar{x}(t) = \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}; \tag{11}$$

The connection of the system (8) with the controller (9) yields the linear delay system:

$$\begin{aligned} \dot{\bar{x}}(t) &= \mathcal{A}_0\bar{x}(t) + \mathcal{A}_1\bar{x}(t - R_0) + \mathcal{B}w(t); \\ z(t) &= \mathcal{C}_0\bar{x}(t) + \mathcal{C}_1\bar{x}(t - R_0) + \mathcal{D}w(t); \end{aligned} \tag{12}$$

where:

$$\mathcal{A}_0 = \begin{bmatrix} A_0 & B_u\hat{C}_0 \\ 0 & \hat{A}_0 \end{bmatrix}; \qquad \mathcal{A}_1 = \begin{bmatrix} A_1 + B_u\hat{D}C_y & B_u\hat{C}_1 \\ \hat{B}C_y & \hat{A}_1 \end{bmatrix};$$

$$\mathcal{B} = \begin{bmatrix} B_w + B_u\hat{D}D_{yw} \\ \hat{B}D_{yw} \end{bmatrix}; \quad \mathcal{C}_0 = \begin{bmatrix} C_z & D_{zu}\hat{C}_0 \end{bmatrix};$$

$$\mathcal{C}_1 = \begin{bmatrix} D_{zu}\hat{D}C_y & D_{zu}\hat{C}_1 \end{bmatrix}; \quad \mathcal{D} = D_{zu}\hat{D}D_{yw};$$

To ensure stability of system (12), the Theorem 4-b in [10] is used. This theorem states that a system such as (12) is asymptotically stable and $\|H_{wz}(s)\|_2^2 < \gamma$, if there exist symmetric and positive definite matrices W, Y_0 and X_j, and matrices F, R, L_j and Q_j, for $j = 0, 1$, such that the following LMIs have a feasible solution:

$$\begin{bmatrix} \mathbf{A}_0 + \mathbf{A}_0^T + X_1 & (\bullet)^T & (\bullet)^T \\ \mathbf{A}_1^T & -X_1 & (\bullet)^T \\ \mathbf{C}_0 & \mathbf{C}_1 & -I \end{bmatrix} < 0 \tag{13}$$

$$\begin{bmatrix} W & (\bullet)^T \\ \mathbf{B} & \mathbf{P}_0 \end{bmatrix} > 0, \quad trace(W) < \gamma \tag{14}$$

where $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{B}$, $\mathbf{C}_0$, $\mathbf{C}_1$, $\mathbf{D}$ and $\mathbf{P}_0$ are given by:

$$\mathbf{A}_0 = \begin{bmatrix} A_0X_0 + B_uL_0 & A_0 \\ Q_0 & Y_0A_0 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} A_1X_0 + B_uL_1 & A_1 \\ Q_1 & Y_0A_1 + FC_y \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} B_w \\ Y_0B_w + FD_{yw} \end{bmatrix}, \quad \mathbf{C}_0 = \begin{bmatrix} C_zX_0 + D_{zu}L_0 & C_z \end{bmatrix},$$

$$\mathbf{C}_1 = \begin{bmatrix} D_{zu}L_1 & 0 \end{bmatrix}, \quad \mathbf{P}_0 = \begin{bmatrix} X_0 & I \\ I & Y_0 \end{bmatrix};$$

This convex problem was numerically solved using the software LMISol [17]. The network parameters were the same used in [8], which are $N = 60$ TCP flows, $R_0 = 0.246$ seconds and $C = 3750$ packets/seconds, that corresponds to a link capacity of 15 Mb/s and average packet size of 500 *bytes*. A feasible solution was found, and, therefore, system (12) is stable.

Once a feasible solution was calculated, the next step is to determine the parameters of controller (9). First, arbitrary singular matrices U_0 and V_0 such that $V_0U_0 = I - Y_0X_0$ should be chosen. The matrices used were $U_0 = X_0$ and $V_0 = X_0^{-1} - Y_0$. Then, the controller parameters are determined by:

$$\begin{bmatrix} \hat{A}_0 & \hat{A}_1 & \hat{B} \\ \hat{C}_0 & \hat{C}_1 & \hat{D} \end{bmatrix} = \mathcal{K}.\mathcal{M}.\mathcal{N}. \tag{15}$$

where:

$$\mathcal{K} = \begin{bmatrix} V_0^{-1} & V_0^{-1}Y_0B_u \\ 0 & I \end{bmatrix};$$

$$\mathcal{M} = \begin{bmatrix} Q_0 - Y_0A_0X_0 & Q_1 - Y_0A_1X_0 & F \\ L_0 & L_1 & 0 \end{bmatrix};$$

$$\mathcal{N} = \begin{bmatrix} U_0^{-1} & 0 & 0 \\ 0 & U_0^{-1} & 0 \\ -C_{y0}X_0U_0^{-1} & -C_{y1}X_0U_0^{-1} & I \end{bmatrix};$$

In the obtained optimal solution, matrices $\hat{A}_1$ and $\hat{C}_1$ are approximately zero, and, therefore, were ignored. As a result, the cancellation of the delay terms in the system lead to a rational controller. The delay cancellation, when possible, is the optimal solution to a H_2 norm minimization problem [10]. The controller $C_{H2}(s)$ represented by its transfer function in the frequency domain is given by:

```
H2-AQM-ProbabilityFunction()
  N     ⇐ NumberActiveFlows();
  p_0   ⇐ 2 * N^2/(R_0 * C)^2;
  q_0   ⇐ N * sqrt(2/p_0) − C * T_p;
  p     ⇐ a * (q − q_0) − b * (q_old − q_0) + c * (p_old − p_0) + p_0;
  p_old ⇐ p;
  q_old ⇐ q;
end;
```

Figure 2: H2-AQM probability calculation algorithm

$$C_{H2}(s) = \frac{0.8243e^{-5}s + 3.186e^{-5}}{0.8978s + 3.4703}; \tag{16}$$

Note that $C_{H2}(s)$ is of order one and not of order two, as expected, since the coefficients of s^2 both in the numerator and in the denominator were ignored, due to their small values.

For a digital implementation of C_{H2}, it is necessary to choose a sampling frequency, f_s, so that a representation in the z-domain is obtained. The frequency chosen was $f_s = 160Hz$, which is the same used in [8]. C_{H2} in the z-domain is, thus, given by:

$$C_{H2}(z) = \frac{az - b}{z - c} = \frac{9.181^{-6}z - 8.962e^{-6}}{z - 0.9761}; \tag{17}$$

The transfer function between $\delta p = p - p_0$ and $\delta q = q - q_0$ presented in (17), can be converted in a difference equation at discrete times kT, where $T = \frac{1}{f_s}$:

$$\delta p(kT) = a\delta q(kT) - b\delta q((k-1)T) + c\delta p((k-1)T); \tag{18}$$

The algorithm for the calculation of the mark/drop probability value in H2-AQM is very simple, and is executed at each sampling instant $1/f_s$ (Figure 2). Initially, the algorithm estimate the number of active TCP flows. Then, the reference values for system stabilization, p_0 and q_0, are computed. Finally, the probability value is calculated based on (18). The algorithm needs two auxiliary variables: q_{old} and p_{old}, which are used to store the values of q and p respectively in the last RTT.

5 Numerical Results

In [8], it was shown that PI-AQM outperformed RED. Therefore, in this section H2-AQM is compared only to PI-AQM. The Simulink software was used to model and simulate both controllers. Congestion control was modelled as a feedback system in which the controller can be chosen to determine the mark/drop probability $p(t)$. Different values of N, C, T_p and R_0 were used in the simulations. The traffic was composed of both TCP and UDP flows. The network parameters used to obtain the equilibrium point were: $N = 60$ TCP flows, $T_p = 0.2$ seconds, $R_0 = 0.246$ seconds and $C = 3750$ packets/seconds. UDP traffic,

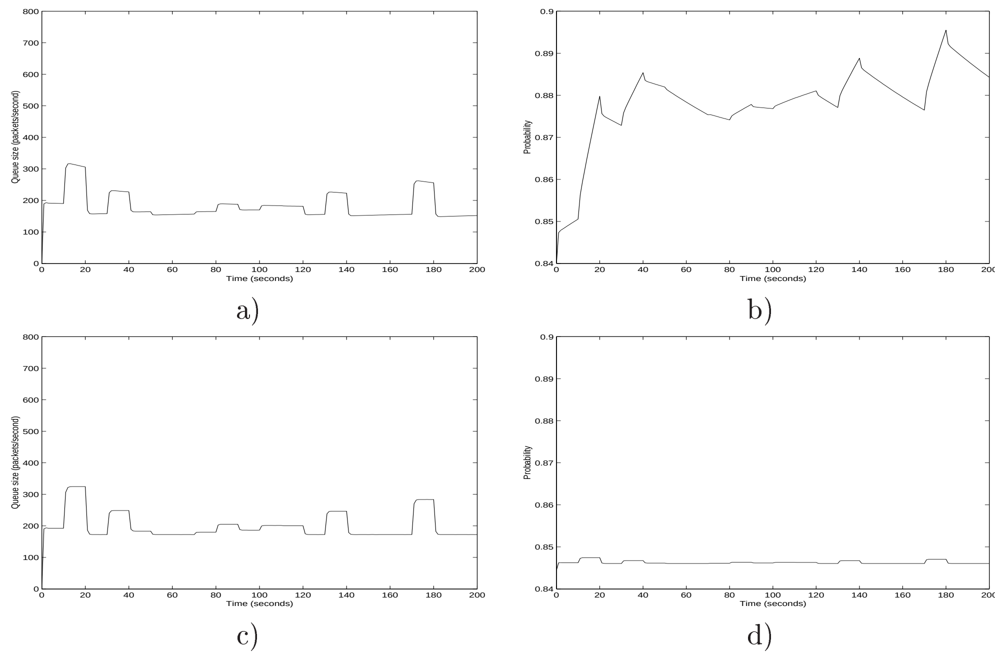

Figure 3: Simulation with $N = 600$ TCP flows, $R_0 = 0.246$ seconds $T_p = 0.2$ seconds and $C = 3750$ packets/second. a) PI-AQM queue b) PI-AQM probability, c) H2-AQM Queue, d) H2-AQM probability

$\omega_q(t)$, is represented by a white noise, and can be up to 20% of the link capacity. The initial condition of the system was defined as $W(0) = 1$ and $q(0) = 0$.

First, it was assessed the impact of the number of active flows on the system (Figure 3). The values of C, R_0 and T_p were kept equal to the initial values and the number of active TCP flows was increased up to 600 flows, leading to a drop/mark probability value of 0.85. It was verified that both controllers stabilize the system even with a high number of flows. However, H2-AQM stabilizes the probability value much faster than does PI-AQM.

The impact of the bandwidth-delay product on the system was also investigated. Firstly, C and N were kept equal to the initial values and the value of T_p was gradually increased and $R_0 = T_p + 0.05$. It was verified that for values of T_p greater than 0.35 seconds, the PI-AQM controller has difficulties to stabilize the system. The queue probability and queue size oscillate considerably. For T_p values higher than 0.55 seconds, PI-AQM does not stabilize the system. The oscillations are wide, which may cause large jitter values. Conversely, H2-AQM produces small oscillations under the same condition, and stabilizes the system (Figure 4).

Secondly, C was varied and T_p and N were kept constant. When the C value was approximately the double of its initial value, PI-AQM start having difficulties to stabilize the queue size and the drop/mark probability. When C value reaches 8000 packets/second, PI-AQM definitely fails to stabilize the system (Figure 5-a and 5b). The oscillations presented are quite intense. Conversely, it can be verified in Figure 5 that H2-AQM

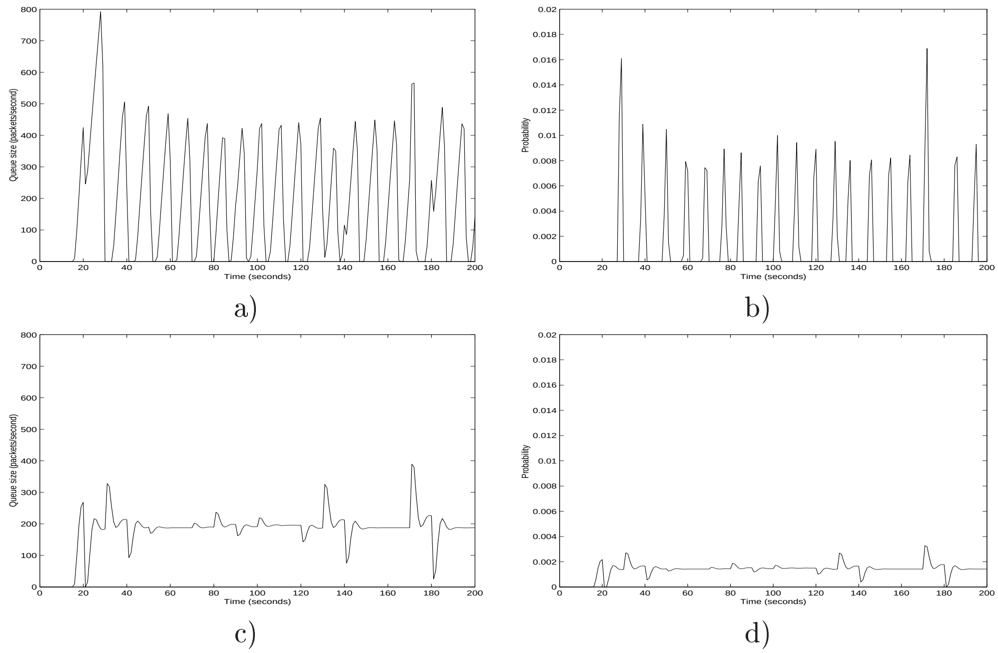

Figure 4: Simulation with $N = 60$ TCP flows, $R_0 = 0.6$ seconds $T_p = 0.55$ seconds and $C = 3750$ packets/second. a) PI-AQM queue b) PI-AQM probability, c) H2-AQM Queue, d) H2-AQM probability

stabilizes the system under the same conditions. Simulation results indicate that PI-AQM does not stabilize the system when the number of active flows is small compared to the bandwidth-delay product, whereas H2-AQM does.

Results obtained in this investigation were somehow expected. PI-AQM uses a simplified plant of the system, and the conditions for which it can stabilize the system are limited. Consequently, PI-AQM does not assure stability for diverse network conditions. On the other hand, H2-AQM provides stability and also produces the optimal drop/mark probability values for different network conditions avoiding unnecessary packet losses.

6 Conclusion

This report introduced an optimal controller for AQM systems fed by both TCP and UDP flows. H2-AQM takes into consideration a detailed description of the system, which assures system stabilization under different network conditions. Although it was used a non-rational approach, the obtained controller was rational, since the cancellation of the delay terms in the system was possible, obtaining the optimal solution to an H_2 norm minimization problem. In addition, the controller is of order one and not of order two, as expected, leading to a simple controller, and consequently, allowing a very simple algorithm. It was shown through simulation that H2-AQM outperforms the PI-AQM policy in achieving AQM objectives. H2-AQM is under implementation in the NS networks simulator, in order to

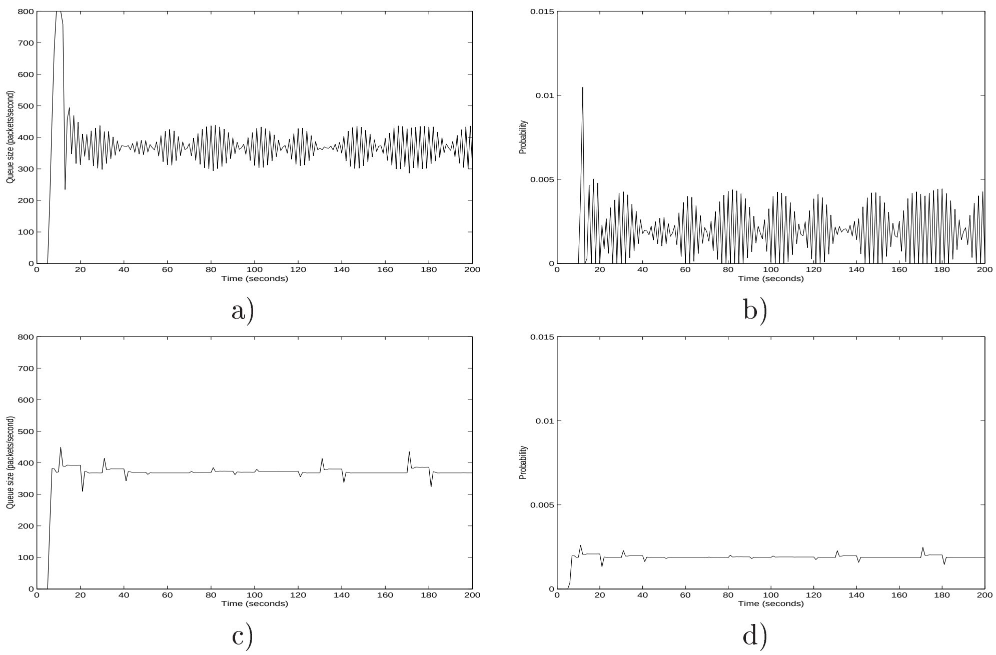

Figure 5: Simulation with $N = 60$ TCP flows, $R_0 = 0.246$ seconds $T_p = 0.2$ seconds and $C = 8000$ packets/second. a) PI-AQM queue b) PI-AQM probability, c) H2-AQM Queue, d) H2-AQM probability

allow the verification of its performance in a dynamic network environment, as well to allow a comparison with other AQM policies.

References

[1] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[2] B. 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.

[3] W. Feng, "Improving Internet congestion control and queue management algorithms," Ph.D. dissertation, University of Michigan, USA, 1999.

[4] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "A self-configuring RED gateway," in *Proc. IEEE INFOCOM'99*, vol. 3, New York, NY, Mar. 1999, pp. 1320–1328.

[5] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proc. IEEE INFOCOM'00*, 2000, pp. 1435–1444.

[6] P. Kuusela, P. Lassila, and J. Virtamo, "Stability of TCP-RED congestion control," in *Proc. ITC-17*, Salvador da Bahia, Brasil, dec 2001, pp. 655–666.

[7] V. Misra, W.-B. Gong, and D. F. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. ACM SIGCOMM'00*, 2000, pp. 151–160.

[8] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," 2002, to appear in special issue of IEEE Trans. Automat. Cont. on Systems and Control Methods for Communication Networks.

[9] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong, "A control theoretic analysis of RED," in *Proc. IEEE INFOCOM'01*, 2001, pp. 1510–1519.

[10] M. C. Oliveira and J. C. Geromel, "Synthesis of Non-Rational controllers for linear delay systems," 2002, to be published.

[11] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[12] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: Utility functions, random losses and ECN marks," in *Proc. IEEE INFOCOM'00*, 2000, pp. 1323–1332.

[13] S. Athuraliya, S. Low, V. Li, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, 2001.

[14] T. J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: Stabilized RED," in *Proc. IEEE INFOCOM'99*, vol. 3, New York, NY, Mar. 1999, pp. 1346–1355.

[15] J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 36, no. 2-3, pp. 203–235, 2001.

[16] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone," in *Proceedings of INET'98*, Geneva, Switzerland, July 1998.

[17] D. P. F. M. C. Oliveira and J. C. Geromel, *LMI Solver User's Guide.* [Online]. Available: http://www.dt.fee.unicamp.br/ mauricio/lmisol/userguide.ps.gz