An Adaptive Control Strategy for AQM Routers Supporting TCP Flows

Yossi Chait* Salomon Oldak† C.V. Hollot‡ Vishal Misra§
University of Massachusetts Amherst MA 01003

January 12, 2001

Abstract

This paper describes the development of an adaptive algorithm for active control management (AQM) routers supporting a varying number of long-lived TCP flows. We motivate the need for adaptation by focusing on the plant's high-frequency gain which is directly related to stability margins and speed of response. We show that the high-frequency gain of the linearized plant is inversely proportional to this number which can vary by at least an order of magnitude. Our adaptive algorithm – the Externally Excited Adaptive Loop (EEAL) – is shown to achieve improved queue level management compared with that achieved using RED and PI based AQMs.

*Corresponding author. Mechanical and Industrial Engineering Department, chait@ecs.umass.edu.
†EET Dept. DeVry Institute of Technology, soldak@socal.devry.edu.
‡Electrical and Computer Engineering Department, hollot@ecs.umass.edu.
§Computer Science Department, misra@cs.umass.edu.

1 Introduction

In the past decade the world has seen an explosion in Internet activity and with it has come increased expectations for performance and services. Internet users now demand faster response and new services such as quality of service (QoS), voice over IP (VoIP) and the bandwidth-intensive video streaming. At the heart of this information exchange is the TCP transport protocol. Under TCP, a sender has authority to set its transmission rate using a window flow-control mechanism. The sender continuously probes the network's available bandwidth and increases its window size to garner maximum share of network resource. For each successful end-to-end packet transmission TCP increases the sender's window size. Conversely, TCP cuts the window in half whenever a sender's packet does not reach the receiver. Such packet losses can affect network performance by decreasing the sender's effective transmission rate and increasing delay due to packet retransmission. By itself, TCP has no information of network mechanisms contributing to packet loss – such as the congested router shown in the simple sender-receiver connection of Figure 1. Thus, routers have been asked to assist in network management by sensing congestion and preemptively signaling TCP rather than have it react to unreceived packets. The simplest form of such active queue management

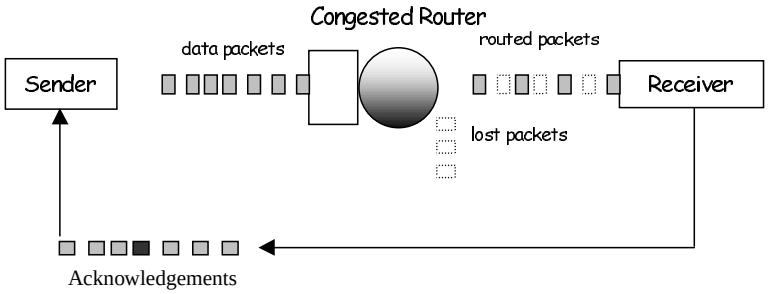

Figure 1: Router congestion results in lost packets. The receiver then signals sender to decrease window size.

(AQM), termed *drop tail*, drops arriving packets when the router's buffer is full. Drawbacks of this scheme include flow-synchronization (see [1]) and performance degradation due to the excessive time-outs and restarts arising when the trailing end of a sequence of data packets is dropped. Motivated by drop-tail's inefficiencies, the random early detection (RED) scheme was introduced in [1]. Rather than waiting for buffer overflow to occur, RED anticipates congestion by measuring the router's queue length and throttling the sender's rate accordingly. Since TCP is an end-to-end protocol, RED achieves this signaling indirectly by randomly marking packets and routing them to the receiver.[1] The receiver, in turn, completes the feedback by acknowledging the receipt of marked packets to the sender; this is depicted in Figure 2 where we emphasize the implicit, delayed,[2] feeding-back of acknowledgment packets. Upon receipt of such acknowledgments, the sender reduces its rate according to the TCP algorithm. The randomness in RED's packet-marking scheme is meant to eliminate flow-synchronization while queue-averaging was introduced to attenuate the effects of bursty traffic on the feedback signal. A crucial drawback in deploying RED stems from tuning

[1] Throughout, we assume use of the Explicit Congestion Notification (ECN) mechanism for marking packets rather than dropping as a way to signal the sender to reduce its window size.

[2] This time delay is equivalent to one round-trip time which is comprised of propagation and queuing delays.

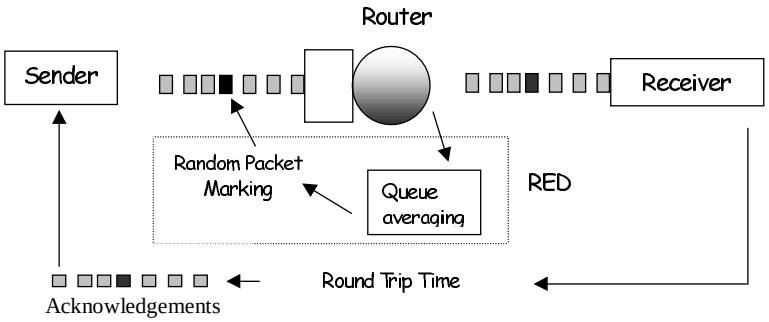

Figure 2: RED randomly marks packets to anticipate congestion.

difficulties;[3] where the performance of detuned RED approaches that of a drop-tail router; e.g., see [2], [3], [4] and [5]. Motivated by these deficiencies in the basic RED mechanism, researchers have proposed modifications as discussed in [4] – [6]. A common observation made amongst these works is that TCP load (the number of TCP flows supported by the router) is a critical parameter in tuning RED. In fact [4], [6] and [7] suggest adaptive schemes in which TCP load is estimated and RED subsequently tuned. Our work is also motivated by a desire to design advanced AQM schemes and explicitly relies on dynamic modeling and feedback control principles.

Recently researchers recognized that AQM schemes are essentially feedback control systems and that their principles can provide critical insight and guidance into the analysis and design of such schemes. While such principles can be found in the study of ATM networks (see for example [8] and [9] and the references cited therein) they have not been applied to TCP-controlled flows. Their absence from the design scene so far is apparently due to a lack of an analytical model of TCP. Fortunately, this roadblock has been recently removed in [10] through the introduction of a fluid-flow model that expresses TCP in a language that allows network control engineers to analyze and design AQM schemes.[4] Indeed, [13], [14] and [15] have accomplished just that by proposing alternative AQM schemes which amount to classical proportional (P) and proportional-integral (PI) feedback controllers.

Congestion control has also been approached from an optimization standpoint in a framework defined by Kelly, Maulloo and Tan [16]. The problem is formulated as a convex program, with the aggregate source utility being maximized subject to capacity constraint. In the primal version of the problem, controllers are designed by taking a penalty function approach to obtain optimal source rates [17], [18]; whereas in a dual formulation [19] controllers are designed to obtain optimal congestion measures (the dual variables). A virtual buffer technique is taken in the primal approach, with the basic idea being to mark packets when a virtual buffer (smaller in capacity and service rate than the actual buffer) overflows. Gibbens and Kelly propose a static virtual buffer configuration [17], whereas Kunniyur and Srikant [18] use an adaptive virtual buffer, adapting the size and capacity of the virtual buffer as a function of the incoming rate to both minimize delay and maximize utilization. Athuraliya and Low [19] design controllers from the duality standpoint, and we note that one version of their REM controller is very similar in flavor to the PI controller proposed in

[3]By tuning we mean selecting the averaging and packet-marking parameters of RED for a given set of network conditions.

[4]We'd also like to point out other recent TCP models in [11] and [12]. They are discussed in [13] in the context of these fluid-flow models.

[13]. The optimization-based approaches largely lead to steady-state equilibria, and do no address the transient performance.

It has been suggested that the load factor has a significant effect on the performance of RED routers (see [4] and [6]). Indeed, in [20], using the same fluid model-based concepts described earlier, we show that stability margins are inversely proportional to the load factor N. When the number of long-lived flows decreases, AQM becomes less aggressive. Conversely, when they increase, stability margins diminish. Load variations of an order of magnitude can significantly affect AQM performance. As suggested in [4] and [6], it may be beneficial to introduce adaptation into AQM to respond to these variations. Let us now review the background behind our specific approach for adaptation.

1.1 Oscillating Adaptive Systems

In this research we explore the use of adaptation in AQM by again borrowing from the area of feedback control. Specifically, we study the utility of the so-called Externally Excited Adaptive Loop (EEAL) algorithm introduced in [21] – [23]). This algorithm appears to be specifically tailored to the type of dynamic variations attributable to variations in the numbers of long-lived flows. This adaptive technique explicitly monitors flow conditions at a core router and dynamically modifies the AQM algorithm to account for these variations.[5]

The EEAL algorithm is rooted in the (related) concepts of dithered feedback systems[6]. Dithered systems are nonlinear systems with self-excited oscillation or externally injected sinusoid of a relatively high frequency. The key idea is that at steady-state both are insensitive to the plant's high-frequency gain. Hence, the loop transmission requirements maybe be smaller than those for an LTI design with same specifications. This concept dates back to the 50's and 60's (e.g., [24]-[26]) where the design was based on the quasilinear conditions from dual-input describing function analysis. Originally the oscillation in these systems was self-generated through the use of an ideal relay in the loop to create a limit cycle[7]. The externally excited adaptive (EEAS) system was able to somewhat relax the lower bound on the oscillation frequency [29] since a high oscillation frequency adds constraint on the loop transmission. Further reduction in the oscillation frequency was achieved by use of an adaptive loop (SOAL) [30]. As reported in [23] the use of an ideal relay in the main loop may significantly degrade the noise response of the system. This led to the development of an EEAL scheme without a nonlinear element [21] which was shown to have distinct advantages over the above formulations. In this paper we make use of this procedure but leave out the justifications for the design steps for lack of space.

The rest of the paper is organized as follows. In the next section we describe the fundamentals of AQM routers from the view point of feedback control. We then motivate the need for adaptation and follow with details of our design of an EEAL algorithm for this problem. Finally, we apply this design to a single queue network and draw comparisons with the PI AQM algorithm.

[5] Specifically, the AQM is automatically tuned to maintain constant loop gain.

[6] we note a difference from the common use of non-adaptive dithering to overcome hard nonlinearities such as friction.

[7] SOAS is related to relay systems were pioneered by Tsypkin [27] and are often the basis of auto tuning algorithms; e.g., see[28].

2 AQM Routers

This section describes the fundamentals of AQM routers. We first present a recently developed nonlinear model for the dynamics of routers supporting TCP flows. We then describe the existing AQM algorithm referred to as RED, and present our recent AQM algorithm – the PI controller.

2.1 A Fluid Flow Model TCP Congestion Control

In [10], a dynamic model of TCP behavior was developed using fluid-flow and stochastic differential equation analysis. We now use a simplified version of that model which ignores the TCP timeout mechanism.[8] This model relates the average value of key network variables and is described by the following coupled, nonlinear differential equations:

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

where $\dot{x}$ denotes the time-derivative of x and

$$\begin{aligned} W &\doteq \text{average TCP window size (packets);} \\ q &\doteq \text{average queue length (packets);} \\ R &\doteq \text{round-trip time} = \frac{q}{C} + T_p \text{ (secs);} \\ C &\doteq \text{queue capacity (packets/sec);} \\ T_p &\doteq \text{propagation delay (secs);} \\ N &\doteq \text{load factor (number of TCP sessions);} \\ p &\doteq \text{probability of packet marking.} \end{aligned}$$

The first term in the $\dot{W}$ equation models the additive increase of window size while the second term captures its multiplicative decrease. We illustrate these differential equations in the feedback control block diagram of Figure 3 which highlights TCP window control and queue dynamics. In the context of these dynamics, an active queue management algorithm amounts to computing a packet-marking probability p as a function of queue length q. This forms, together with (1), a feedback control system as illustrated in Figure 4.

2.2 The RED and PI AQM Algorithms

In [13], [14] and [15], linearization techniques and feedback control principles are used to analyze and design AQM algorithms. The AQM problem is posed as one of feedback control as illustrated in Figure 4 where the transfer-function $P(s)$ denotes the linearization of the TCP dynamics (1) around an operating point (W_0, p_0, q_0) and $C(s)$ represents an AQM control law. The *plant* transfer function $P(s)$ is given by

$$P(s) = \frac{(\frac{C^2}{2N})e^{-sR_0}}{(s + \frac{2N}{R_0^2 C})(s + \frac{1}{R_0})} \tag{2}$$

[8] It is believed that with implementation of SACK and ECN, the effect of timeouts will be minimized

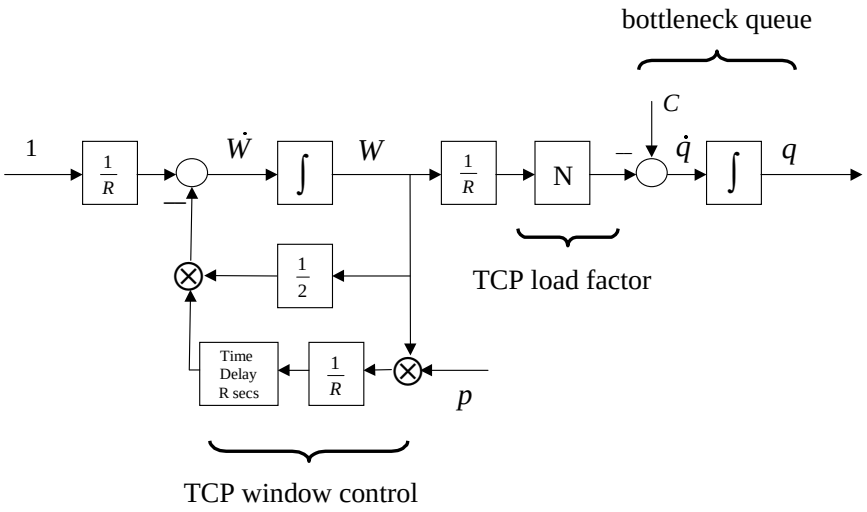

Figure 3: Block-diagram of TCP's congestion-avoidance flow-control mode.

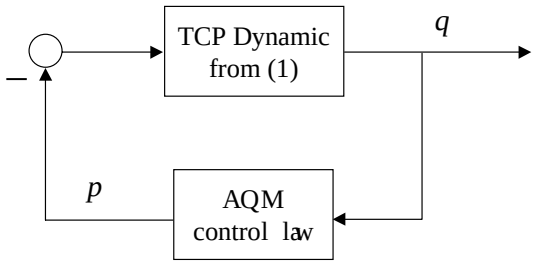

Figure 4: AQM as feedback control.

which explicitly relates the network parameters N (TCP load), R_0 (round-trip time) and C (queue capacity) to the dynamics of AQM. These relationships are discussed in detail in [13] and [14]. We would like to point out here that the *gain* of the plant $\frac{C^2}{2N}$ and the delay element e^{-sR_0} play crucial roles in the stability and performance of AQM. Roughly speaking, stability margins decrease with smaller load N, larger capacity C or larger round-trip time R_0.

By and large, present AQM algorithms are based on RED. However, recently, [10] analyzed the RED algorithm from a control view point and [15] suggested choices for designing such a PI controller which stabilizes the linearized feedback system and offers robustness to given ranges of TCP loads and round-trip times. The conclusion in [15], supported by simulations, is that PI control achieves significant performance improvement compared with RED-based AQM. To illustrate this key finding, consider a simple network topology consisting of a single bottleneck queue and a set of long-lived TCP flows. The queue has bandwidth capacity of $C = 3500$ packets/sec, the number of flows is $N = 400$, the and the desired queue buffer's level is $q_0 = 175$ packets. The controller is given by

$$C(s) = 9.6426(10)^{-6}\frac{(\frac{s}{0.53}+1)}{s}.$$

The `NS` [31] simulation results in Figure 5 show that the PI controller is capable of regulating the queue's length to $q_0 = 175$ while RED loses control allowing the queue to be locked in a persistent overflow condition. At such load levels, the loss probability has become so high that RED has forced the queue length beyond the buffer size.

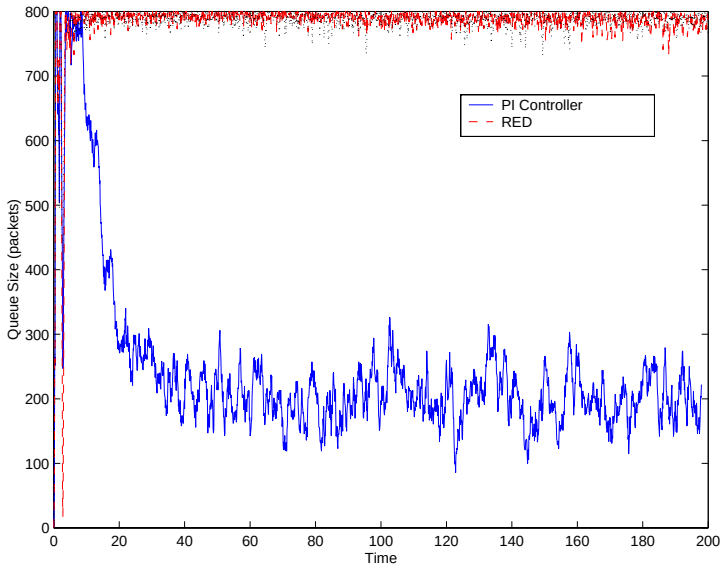

Figure 5: Simulations under high TCP load. RED overflows while PI regulates.

Another benefit of PI control is its ability to regulate queuing delay independent of TCP load. An important consideration in designing AQM systems is the tradeoff between queuing delay and utilization. Intuitively, larger buffers lead to higher link utilization, but they also result in larger queuing delays. With the PI controller, the delay is essentially tunable with a single parameter q_0. Larger values of q_0 give larger delay and utilization. In contrast, with RED, the delay is a function

network conditions such as load level and packet-marking profile parameters min_{th}, max_{th} and p_{max}.

In the above, we have motivated the need for PI-type control algorithms in AQM routers using results from [13]- [15]. In the sequel, we first show that the load factor N has a significant effect on the system performance. We then present an adaptive algorithm that, coupled with a PI-type design, can adapt this controller's gain to reflect dynamic changes in N.

3 The Adaptive AQM

In this section we describe the salient features of our adaptive controller leading to the comparison with the PI algorithm presented in the previous section. We first motivate the need for adaptation.

3.1 Why Adaptive?

As described earlier, the open-loop transfer function from the probability p to the queue level q derived at the linearized model (at equilibrium) for a single-queue TCP model is given by (2)

$$P(s) = \frac{\frac{R_0 C^2}{2N^2}}{s + \frac{2N}{R_0^2 C}} \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}}$$

Two asymptotic quantities, the plant's high-frequency gain and low-frequency gain are important measures of the system's performance. The plant's low-frequency gain (i.e., steady-state or DC gain) is given by

$$P(s) \longrightarrow \frac{(R_0 C)^3}{(2N)^2}, \quad s \to 0,$$

while the plant's high-frequency gain is

$$P(s) \longrightarrow \frac{\frac{C^2}{2N}}{s^2} \doteq \frac{K_{hf}}{s^2}, \quad s \to \infty.$$

Putting aside phase variations, if K_{hf} is not fixed then in order to insure a robust gain margin, control design must be executed with respect to the worst-case gain, that is $\max[K_{hf}]$. If the controller is fixed, as is the case in linear, time-invariant (LTI) systems, such a robust design introduces a limitation on the achievable performance. To see this, consider the case where the load lies in the range $N \in [N_{min}, N_{max}]$. Given that the fixed robust controller was designed using N_{max}, if $N = N_{min}$ then $K_{hf} =$ is $\frac{N_{min}}{N_{max}}$ smaller. A lower high-frequency gain implies a lower system bandwidth. And lower bandwidth implies slower system in terms of rise and settling time. Given that $\frac{N_{min}}{N_{max}}$ can be as low as 0.1, it is apparent that we should investigate feasibility of adaptive controllers. One such controller is described below using an Externally Excited Adaptive Loop (EEAL) scheme.

3.2 The Adaptive Control Problem

The block diagram of an EEAL system is shown in Figure 6. Note that P_h denotes the fixed plant in Eq. (2) evaluated at a nominal value of the load N. The adaptive algorithm design involves this procedure from [21]: define the nominal closed-loop performance and then design a linear controller

G, compute a lower bound of the oscillation frequency ω_0 and then design of the linear filters G_2G_{3h} and G_1, and finally, design the adaptive loop. Let us now proceed with details of our design.

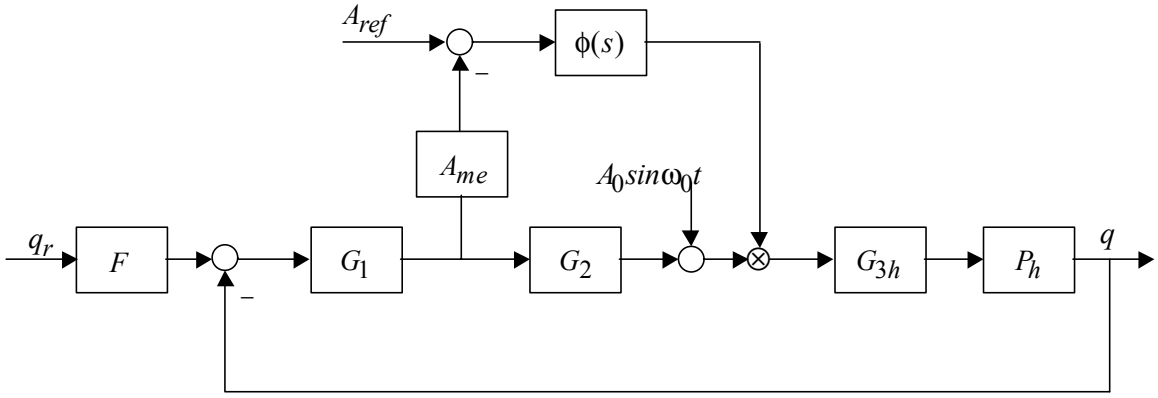

Figure 6: Block diagram of an EEAL system.

3.2.1 Specifications

As a target for the closed-loop transfer function from q_{ref} to q we use a nominal load level of $N = 60$ (i.e., $P_{60}(s)$ in [15]) and the controller in [15] as an indication for achievable performance. While more detailed performance specifications can be defined and used in conjunction with recent robust control techniques, this is not needed here. The large time delay in this system and the lack of disturbance models allow for use of straightforward frequency response control design. The desired closed-loop model is given by

$$T(s) = \frac{1125 \exp^{-0.45s}}{(s^2 + 1.5s + 2.25)(s + 5)(s + 10)^2}.$$

Note that we use a larger delay than the steady-state value since during transients the queue level is higher causing larger delays (the need for such conservatism is yet to be proved). The remaining specifications are (using notation in [21]):

- $R_e = \frac{175}{s}$ the "extreme" command input (i.e., $q_{ref} = 175$),
- $m = 3$ maximal oscillation amplitude at the plant output,
- $A = 1$ (typical) oscillation amplitude at the measurement point, and
- $KK_3 = 1$ arbitrary choice.

3.2.2 Nominal LTI Design

Again, the nominal plant is the $P_{60}(s)$ in [15] with a worse-case delay (0.45 instead of 0.25 seconds)

$$P(s) = \frac{117126K \exp^{-0.45s}}{(s + 0.53)(s + 4.1)} \doteq KP_h(s); \quad 0.1 \leq K \leq 1$$

A standard QFT design ([32]) is carried out assuming $K = 1$ (i.e., K has no uncertainty). We use a $6dB$ margin constraint (i.e., peaking in the complimentary sensitivity $|T(j\omega)| \leq 2$) to guide the loopshaping procedure as shown in Figure 7. The following controller was designed:

$$G(s) = \frac{15.51(s + 1.75)(s^2 + 6.52s + 16.03)}{s(s + 20.17)(s + 33.7)(s + 74.38)(s + 214.1)}.$$

The nominal loop is

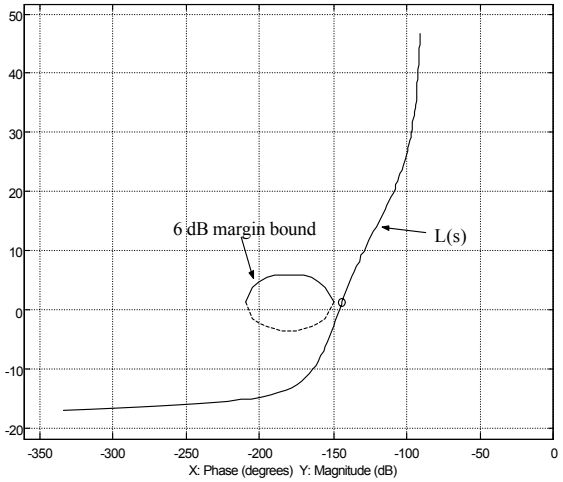

Figure 7: The nominal loop $L(s)$ on a Nichols chart and a $6dB$ margin bound.

$$L(s) \doteq P(s)G(s) = \frac{1816624.26(s + 1.75)(s^2 + 6.52s + 16.03)\exp^{-0.45s}}{s(s + 0.53)(s + 4.1)(s + 20.17)(s + 33.7)(s + 74.38)(s + 214.1)}.$$

In what follows, the design of the adaptive scheme was carried out using [21]; additional insight can be found in [22]-[23].[9]

3.2.3 Choice of the Oscillation Frequency

The lower bound of the oscillation frequency ω_0 is given by Eq. (6.11) in [21] (page 112)

$$\begin{aligned} |L(j\omega)| &\geq \frac{3}{m}\left|R_e(j\omega)T(j\omega)(j\omega + p)\right|, \quad \omega \geq \omega_0 \\ &\approx \frac{3\omega_0}{m}\left|R_e(j\omega_0)T(j\omega_0)\right|, \quad \omega \geq \omega_0, \end{aligned} \tag{3}$$

where we have used $p \leq \frac{\omega_0}{3}$. Both sides of (3) are plotted in Figure 3 with the choice of $p = 6$. . It can be observed that the inequality (3) is satisfied for $\omega_0 > 18$ rad/sec. We select $\omega_0 = 20$ substantiating our earlier choice of $p = 6$.

[9] We note that in this paper we assume a fixed $P_h(s)$ which is not the case. While simulations indicate that uncertainty in $P_h(s)$ does not affect EEAL performance, further study is required.

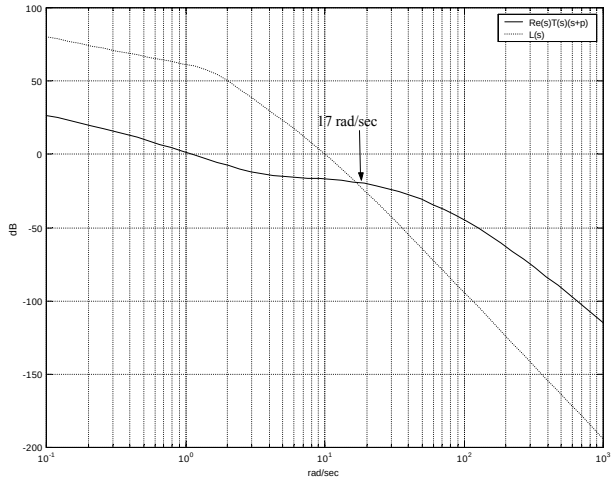

Figure 8: Verification of (3) with $p = 6$.

3.2.4 Design of Fixed Filters in the Main Loop

Using Eq. (6.12) in [21] (page 115) with $\alpha = 3$ and $p = 6$

$$G_2 G_{3h} = \frac{\alpha R_e(s) T(s)(s+p)}{AKK_3 P_h(s)\psi(s)} = \frac{5.04(s+0.53)(s+4.1)(s+6)}{s(s^2+1.5s+2.25)(s+5)(s+10)^2\psi(s)} \tag{4}$$

Let the relative degree in $\psi(s)$ be so that the relative degree of $G_2 G_{3h}$ is 1. We select

$$\psi(s) = \frac{80^2}{(s+80)^2}$$

with the corner frequency two octaves above ω_0. $G_2 G_{3h}$ then reduces to

$$G_2 G_{3h} = \frac{0.000787(s+0.53)(s+4.1)(s+6)(s+80)^2}{s(s^2+1.5s+2.25)(s+5)(s+10)^2}.$$

Let us take $G_2(s) = 1$. Using model order reduction in [32], we obtain

$$G_{3h} = \frac{0.000787(s+0.53)(s+4.53)(s+80)^2}{s(s^2+1.5s+2.25)(s+7.98)(s+11.54)}.$$

If necessary, G_2 can be used to reduce the dither amplitude A_0 (computed at the end of this section).

Next, using Eq. (6.13) in [21] (page 115)

$$\begin{aligned} G_1(s) &= \frac{AL(s)\psi(s)}{\alpha R_e(s) T(s)(s+p)} \\ &= \frac{19678.56(s+1.75)(s+5)(s+10)^2(s^2+1.5s+2.25)(s^2+6.52s+16.03)}{(s+0.53)(s+4.1)(s+6)(s+20.17)(s+33.7)(s+74.38)(s+80)^2(s+214.1)}. \end{aligned}$$

The pre-filter F(s) is found from

$$F(s) = \frac{1 + L(s)}{L(s)} T(s)$$

Using [33], a rational approximation (see Figure 9) is found

$$F(s) = \frac{90.97(s + .49)(s + 23.95)}{(s + 0.39)(s^2 + 2.17s + 6.37)(s^2 + 39.33s + 424.3)}$$

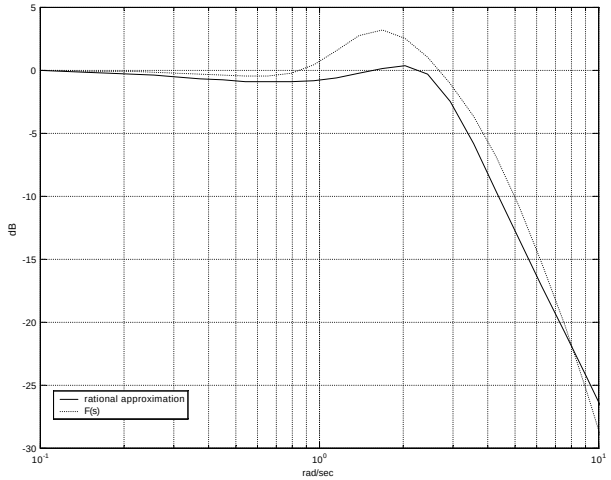

Figure 9: Rational approximation of $F(s)$ (solid - approximation) .

Finally, the amplitude of the injected sinusoid A_0 is found from [21]

$$A_0 = \left| \frac{A(1 + L(j\omega_0))}{L(j\omega_0)} G_2(j\omega_0) \right| = 11.04$$

3.2.5 Design of the Adaptive Loop

The linearized and approximated closed-loop transmission for the adaptive loop is given by Eq. (6.14) in [21] (page 118). The effective open-loop transmission is (we have added a delay of 0.45 not shown in [21])

$$L_a(s) = A_{me}(s)\Phi(s)K \left| G_1(j\omega_0) G_{3h}(j\omega_0) p_h(j\omega_0) \right| t_{cs2}(s) \exp^{-0.45s} \tag{5}$$

where $\Phi(s)$ is design as a stabilizing type 1 (to achieve zero steady-state error) filter. In addition, for fast adaptation, it should have largest possible bandwidth with reasonable margins. From empirical studies [21] it was found that a reasonable choice of T_{cs2} is

$$t_{cs2}(s) = \frac{\left(\frac{\omega_0}{2}\right)^2}{s^2 + 1.4\left(\frac{\omega_0}{2}\right) s + \left(\frac{\omega_0}{2}\right)^2} = \frac{400}{s^2 + 28s + 400}$$

The filter A_{me} represents the dynamics of the peak detector. For pure sinusoidal signals, a detector made of a rectifier operator followed by a low-pass $\frac{1}{s+1}$[10]. Given the high levels of noise in our system, we designed a band-pass filter as the first signal processing step in the detector (see [21]). This filter with a center frequency of 20 Hz and a 1 Hz band is given by

$$G_{BP}(s) = \frac{s^2}{(s^2 + 6.83s + 245.86)(s^2 + 11s + 630.07)}$$

The overall dynamics of A_{me} must take into account the dynamics of the band-pass filter when excited by the dither signal. An approximation for this was found to be

$$A_{me} = \frac{10}{s + 10}$$

Taking $K = 1 = K_{max}$, we design $\Phi(s)$ using QFT (note $|G_1(j\omega_0)G_{3h}(j\omega_0)p_h(j\omega_0)| = 0.098$). Again, we use a $6dB$ margin constraint to guide the loopshaping of the open-loop (5). The filter is given by (see Figure 10)

$$\Phi(s) = \frac{1}{s}$$

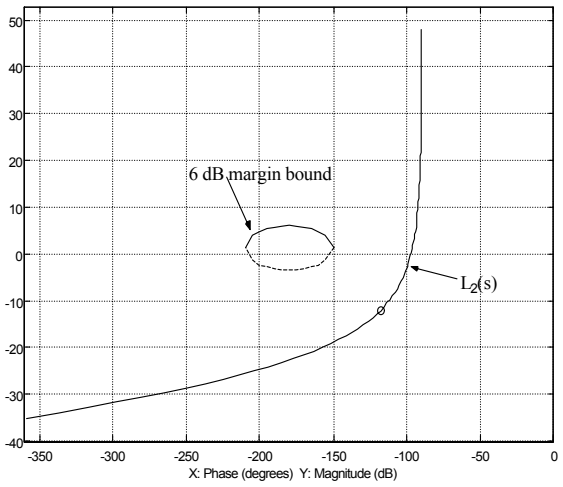

Figure 10: Design of $\phi(s)$.

3.2.6 Extensions

In [21], it is strongly recommended that during start up time, the adaptation be allowed to converge before external inputs are applied. This may be too restrictive for TCP congestion control applications. Based on a detailed experimental study of the adaptive scheme applied to our problem, we were able to overcome this limitation. Specifically, it was found that the adaptation error and the integrator in the adaptation loop must be properly constrained.

[10] Such detector is also used in relay control to compute the amplitude of the limit cycle.

4 Single-Queue Network Application

In this section we compare the performance of the LTI controller similar to the one in [15] with the adaptive scheme described above. Our model includes a number of saturations: queue buffer $q \in [0, 800]$, TCP window $W \in [1, 50]$ and loss probability $p \in [0, 1]$. In addition, $T_p = 0.2$ sec and the reference (i.e., desired) queue level was set to $q_{ref} = 175$. We use the following model to describe the bounded, slow-varying noisy load N

$$N = 100 + \text{white noise}$$

where the white noise is a Simulink band-limited white noise with noise power of 10000 and Sample Time of 10. Queue levels under LTI and EEAL control are compared in Figures (12)-(13).

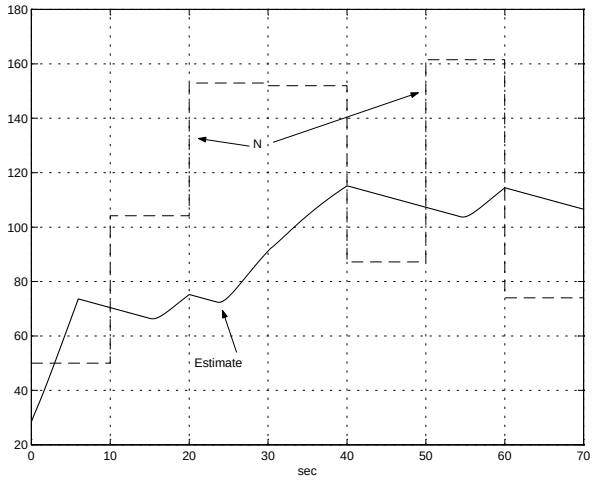

Figure 11: Stochastic load and its EEAL estimation.

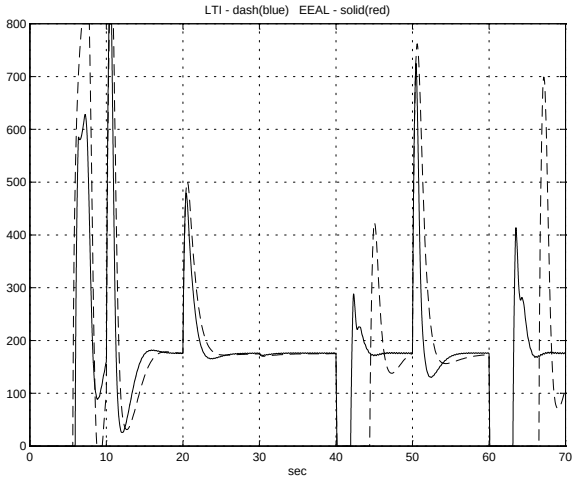

Figure 12: Long duration queue level comparison with stochastic load variation in Figure 11.

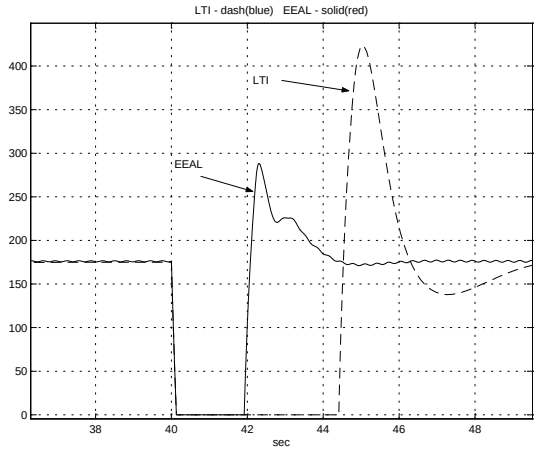

Figure 13: Zoomed Figure 12 showing faster recovery from an empty queue level with EEAL scheme.

An additional test was run to compare the ability of the controller to escape a fully congested router brought about by a sudden increase in the load N at $t = 30$ from 100 to 400 as shown in Figure 14. This step change results in the loop's bandwidth being cut by a factor of 4. While the LTI design remains fixed, that is, 4 times more sluggish, the EEAL scheme is able to increase the loop gain resulting in a faster response. The clear improvement in the queue level seen in Figure 15 occurs even with the adaptation reaching only 50% of its steady-state value.

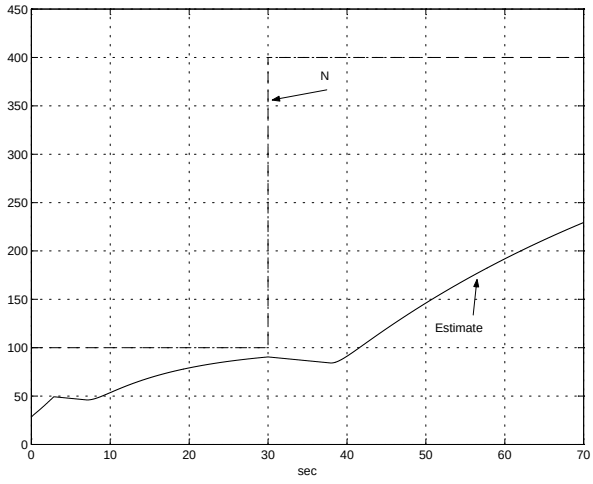

Figure 14: Pulse load variation and its EEAL estimation.

5 Conclusions

We have motivated the need for adaptation in AQM routers due to the time-varying nature of the network load. We used the EEAL algorithm to implement an adaptive AQM and demonstrated

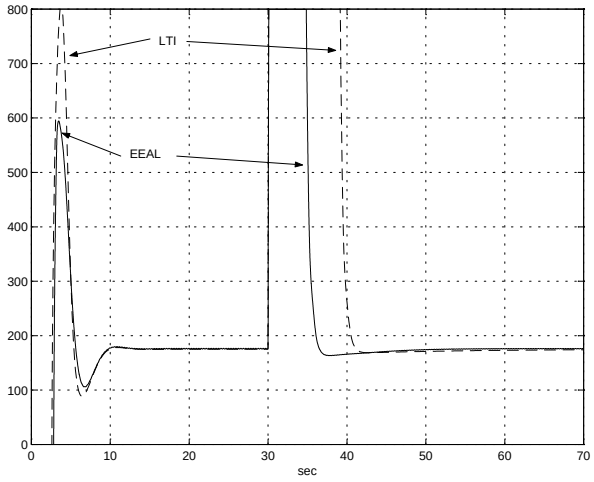

Figure 15: Pulse load variation and its EEAL estimation.

its advantages when applied to a single-queue network. In our ongoing research, we are testing the scalability of this concept in a multi-queue network and are studying the effect of noise on the adaptation loop.

References

[1] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, August 1997.

[2] M. Christiansen, K. Jeffay, D. Ott, and F. Smith, "Tuning Red for web traffic," in *Proceedings of ACM/SIGCOMM*, 2000.

[3] M. May, T. Bonald, and J.-C. Bolot, "Analytic evaluation of RED Performance," in *Proceedings of INFOCOM 2000*.

[4] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "A self-configuring RED gateway," in *Proceedings of INFOCOM 1999*.

[5] M. May, C. Diot, B. Lyles, and J. Bolot, "Influence of Active Queue Management Parameters on Aggregate Traffic Performance." Work in progress. ftp://ftp.sprintlabs.com/diot/aqm.zip.

[6] T. J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: Stabilized RED," in *Proceedings of INFOCOM 1999*.

[7] D. Lin and R. Morris, "Dynamics of random early detection," in *Proceedings of ACM/SIGCOMM*, 1997.

[8] S. L. Y. Zhao and S. Sigarto, "A linear dynamic model for design of stable explicit-rate ABR control schemes," in *Proceedings of IEEE Infocom'97*, 1997.

[9] T. B. E. Altman and R. Srikant, "Robust rate control for ABR sources," in *Proceedings of IEEE INFOCOM'98*, 1998.

[10] 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*, 2000.

[11] S. Mascolo, "Congestion control in high-speed communication networks," *Automatica*, vol. 35, pp. 1921–1935, March 1999.

[12] F. Kelly, "Mathematical modeling of the Internet," in *Mathematics Unlimited - 2001 and Beyond*, 2000.

[13] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "Analysis and design of controllers for AQM routers supporting tcp flows." submitted to special issue of *IEEE Transactions on Automatic Control* on "Systems and Control Methods for Communication Networks".

[14] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "A control theoretic analysis of red." to appear in *Proceedings of INFOCOM 2001*, also available at ftp://gaia.cs.umass.edu/pub/Misra00-RED-Control.ps.gz.

[15] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for aqm routers supporting tcp flows." to appear in *Proceedings of INFOCOM 2001*, also available at ftp://gaia.cs.umass.edu/pub/Misra00-AQM-Controller.ps.gz.

[16] F. P. 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.

[17] R. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, 1999.

[18] S. Kunniyur and R. Srikant, "A Time Scale Decomposition Approach to Adaptive ECN Marking," to appear in *Proceedings of Infocom 2001*, April 2001.

[19] S. Athuraliya and S. Low, " Optimization flow control, II: Random Exponential Marking." Submitted for publication, `http://www.ee.mu.oz.au/staff/slow/research/`, May 2000.

[20] Y. Chait, C.V. Hollot, V. Misra, S. Oldak, D. Towsley and Wei-Bo Gong, "Fluid model-based controllers for AQM routers supporting TCP flows," *ACC 2001*, submitted.

[21] Salomon Oldak, *Synthesis Theory of Dithered Feedback Systems*, Ph.D. Thesis, Department of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel, 1991.

[22] Horowitz I., Oldak S., Shapiro A., "Extensions of Dithered Feedback Systems," *Int. J. Control*, 1991, Vol. 54, No. 1, 83-109.

[23] Oldak S., Horowitz I., Shapiro, A., "The Sensor Noise Problem in Dithered Feedback Systems,", *Automatica*, 1992, Vol. 28, No. 5, 1021-1026.

[24] Smyth, R.K., and Nahi, N.E., "Phase and amplitude sinusoidal dither adaptive control system," *IEEE Trans. Automatic Control*, 1963, 311-321.

[25] Horowitz, I., "Comparison of linear feedback systems with self oscillatory adaptive systems," *IEEE Trans. Automatic Control*, 1964, AC-9, 386-392.

[26] Boskovith, B., and Kauffman, R.E., "Evolution of the Honeywell first generation adaptive autopilot and its application to F094, F101, X15 and X20 vehichles," *J. Aircraft*, 1966, Vol. 3(4).

[27] Tsypkin, Y.Z., *Relay control systems*, 1984, University press, Cambridge, UK.

[28] K.J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1989.

[29] Horowitz, I., Smay, J.W., and Shapiro, A., "A synthesis theory for the externally extited adaptive system (EEAS)," *IEEE TAC*, 1974, Vol. 19(2), pp. 101-107.

[30] Horowitz, I., and Shapiro, A., "A synthesis theory for multiple-loop oscillating adaptive systems," *Int. J. Control*, 1979, Vol. 29(6), pp. 963-979.

[31] NS Simulator, Version 2.1b5, available from `http://www-mash.cs.berkeley.edu/ns`.

[32] C. Borghesani, Y. Chait and O. Yaniv, *The Quantitative Feedback Theory Control DesignToolbox*, for use with MATLAB, the MathWorks Inc., Natick, MA, 1994.

[33] *Frequency Domain System Identification Toolbox*, for use with MATLAB, the MathWorks Inc., Natick, MA, 1994.