Fixed and Adaptive Model-Based Controllers for Active Queue Management[1]

Yossi Chait[2] C.V. Hollot[3] Vishal Misra[4]
Salomon Oldak[2] Don Towsley[4] Wei-Bo Gong[3]
[3]MIE [4]ECE [5]CS
University of Massachusetts, Amherst MA 01003
{chait,hollot,gong}@ecs.umass.edu, {misra,towsley}@cs.umass.edu

Abstract

In this paper we present new model-based controllers for active queue management (AQM) supporting TCP flows. Our control design is based on linearization of a previously developed TCP model using fluid flow and stochastic differential equations. An AQM using a simple PI controller is shown to significantly outperform RED AQM. However, we show that both AQMs are unable to maintain performance as the number of flows increases. To overcome this, we introduce a special type of adaptation, the externally excited adaptive loop (EEAL) controller. In comparison to the PI AQM, the EEAL AQM exhibits faster response to flow changes, and is able to give a dynamic estimate of the number of flows.

1 Introduction

Active Queue Management (AQM) is a very active research area in networking. Specifically, the RED [1] variant of AQM has generated extensive interest in the community. Understanding the behavior of RED has largely remained a "simulate and observe" exercise, and tuning of RED has proven to be a difficult job. Numerous variants have been proposed [2], [3], [4] to work around some of the performance problems observed with RED. In [5], we performed a control theoretic analysis of a linearized model of TCP and RED obtained using fluid flow and stochastic differential equations . The analysis enabled us to present design guidelines for RED, which we verified via simulations using `ns` [6]. Our investigation revealed two limitations of RED. The first limitation deals with the tradeoff between speed of response and stability. An aggressive RED design which resulted in fast response time was found to have relatively low stability margins, while a more stable RED design exhibited very sluggish responses. The other limitation of RED is the coupling between queue length and loss probability. This coupling results in a control system that has steady-state regulation errors. In this paper we present two new model-based controllers, the first being a Proportional-Integral (PI) controller. Non-linear simulations using `ns` [6] show PI AQM to be a robust and outperform RED AQM under almost all scenarios considered. We then introduce an adaptive controller to overcome the limited ability of of fixed controllers to account for variations in the process dynamics. Specifically, as the number of TCP flows increases, the open-loop dynamics become more sluggish and fixed-gain PI controllers are unable to speed up the closed-loop response. Specifically, we show that the number of TCP flows is related to the high-frequency gain of the linearized model and then employ a special form of an adaptive controller to compensate for such variations. Using simulations, we show that the adaptive AQM outperforms the PI AQM in all scenarios considered.

The rest of the paper is organized as follows. In Section 2, we present the linearized control system developed in [5]. Section 3 presents simulations using the

[1]This work is supported in part by the National Science Foundation under Grant ANI-9873328 and by DARPA under Contract DOD F30602-00-0554.

PI AQM and also compares it's performance with the RED AQM. In Section 4 we present a comparison between the PI and the adaptive controllers. Finally we present our conclusions in Section 5.

2 Background

In [5], we linearized a non-linear dynamic model for TCP/AQM developed in [7]. The block diagram of the non-linear model is shown in Figure 1, while the linearized model is depicted in Figure 2; see [5] for linearization details.

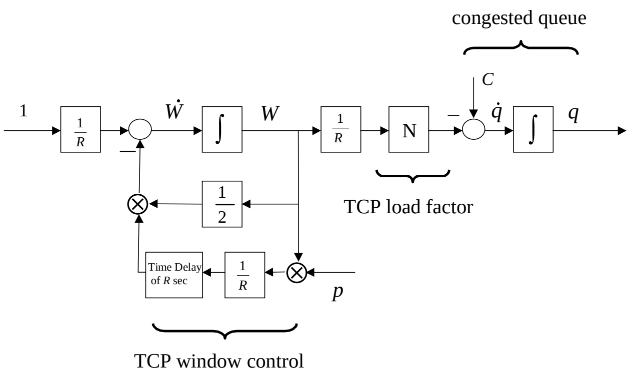

Figure 1: Block-diagram of a TCP connection.

In Figure (2), $G(s)$ is the controller, $P(s)$ is the "plant" or TCP/AQM system we are trying to control, e^{-sR_0} is the delay due to the round trip time R_0. Specifically, the plant $P(s)$ is the product $P_{tcp}(s)P_{queue}(s)$ where

$$
\begin{aligned}
P(s) &\doteq P_{tcp}(s)P_{queue}(s) \\
P_{tcp}(s) &= \frac{\frac{R_0C^2}{2N^2}}{s+\frac{2N}{R_0^2C}}; \\
P_{queue}(s) &= \frac{\frac{N}{R_0}}{s+\frac{1}{R_0}}.
\end{aligned}
$$

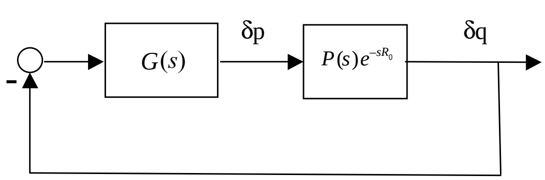

Figure 2: Block diagram of a linearized AQM control system

where

$$
\begin{aligned}
R &\doteq \text{round-trip time at the operating point (sec)} \\
C &\doteq \text{queue capacity (packets/sec)} \\
N &\doteq \text{load factor (number of TCP sessions)}
\end{aligned}
$$

The AQM scheme studied in [5] used the well-known RED controller [1]. RED consists of a low-pass filter (LPF) and nonlinear gain map as shown in Figure 3. The form of the LPF was derived in [7]. The pole K is equal to $\log_e(1-\alpha)/\delta$, where α is the averaging weight and δ is the sampling frequency. Normally RED updates its moving average on every packet arrival, and hence δ is $1/C$. At high load factors N this sampling frequency exceeds C, whereas at low load factors it falls below C. On an average however, under the assumption of a stable congested queue, the sampling frequency is C. The output of the RED controller is a loss probability as a function of the average queue length, as depicted in the RED profile in Figure 3. This loss probability is utilized in dropping or marking packets.

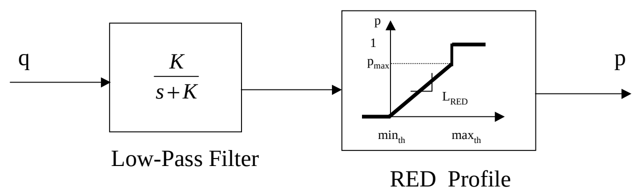

Figure 3: RED as a cascade of low-pass filter and non-linear gain element.

A transfer-function model for RED is:

$$G(s) = G_{red}(s) = \frac{L_{red}}{s/K+1},$$

where

$$L_{red} = \frac{p_{max}}{max_{th} - min_{th}}; \quad K = \frac{log_e(1-\alpha)}{\delta}.$$

Based on the linearized model, we gave design rules in [5] for obtaining a stable linear feedback control system with the RED controller. More importantly, with the new model for this process we are able to design and analyze new model-based controllers with significant performance improvements over the RED design.

3 The PI AQM

A crucial limitation in RED design is that the response time of the control system is sluggish. Specifically, the response time of the system is limited to $1/\omega_g$ sec, where $\omega_g = 0.1 \min\{p_{tcp}, p_{queue}\}$. The multiplication factor of 0.1 is the tradeoff between stability margins and speed of response. Larger values than 0.1 yields more responsive designs, however they have lower stability margins. Intuitively speaking, the lag introduced by the low pass filter is a cause of the sluggishness of the response. One way to improve the response time of the system is to remove the LPF, and introduce classical proportional controller. In proportional control, the feedback signal is simply the regulated output (queue length) multiplied by a gain factor. In the RED context, it corresponds to obtaining the loss probability from the instantaneous queue length instead of the averaged queue length. While we appreciate that one of the design goals of the LPF was to dampen out bursty flows, from a control standpoint such averaging in a feedback system can lead to instability and low frequency oscillations in the window control. Nevertheless, we do not recommend that the proportional controller replace the LPF mechanism in RED, rather, we propose a classical PI controller

$$G_{PI}(s) = K_{PI}\frac{(\frac{s}{z}+1)}{s}.$$

To validate the performance of the PI AQM, we implemented it in `ns` with a sampling frequency of 160 Hz. q_{ref} for the PI controller was chosen to be 200 packets. We ran several experiments as follows (see [8] for details).

In the first experiment we consider a queue with 60 FTP (greedy) flows, and 180 HTTP sessions. The link bandwidth is 15 Mb/s, and the propagation delays for the flows range uniformly between 160 and 240 ms, with average packet size being 500 Bytes. The buffer size is 800 packets. We also provide some time-varying dynamics to compare the speed of response of the RED AQM vs. the PI AQM. At time $t = 100$, 20 of the greedy flows drop out, and at time $t = 140$ they start back again. We set the slope of the loss profile to be the gain calculated in the example above, varying the loss linearly from 0 at queue length 100 with the slope specified by gain. Note that the buffer size of 800 puts an upper limit on the marking probability giving $P_{max} = (800-100)\cdot L_{red} \approx 0.04$. For the traditional RED controller with an LPF, we use the parameters derived for stable operations in Example 2 of [5], with $min_{th} = 150$ and $max_{th} = 600$, and an averaging weight of $1.33e^{-6}$. The queue length plots for the two AQMs are depicted in Figure 4. The faster response time as well as the regulation of the output to a constant value by the PI AQM is clearly observed. The PI AQM controller is largely insensitive to the load factor variations and attempts to regulate the queue length to the same value of 200 packets (with an average size of 500 Bytes).

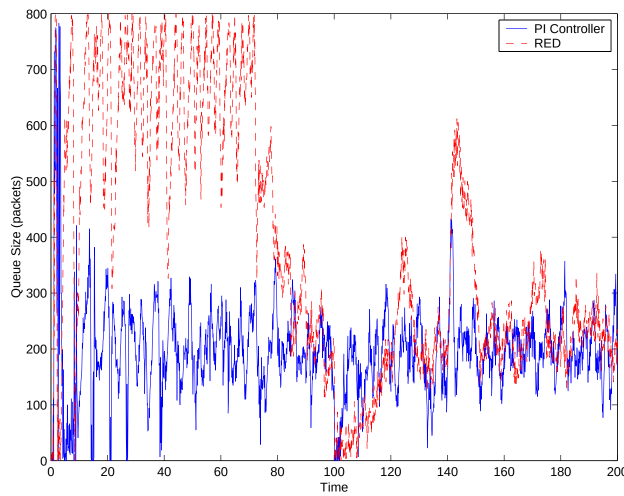

Figure 4: Experiment 1

In the second experiment we test the controllers at the other end of the stability spectrum by reducing the number of FTP flows to 16. As observed in Figure 5, the RED AQM exhibits oscillations while the PI AQM operates in a relatively stable mode.

In the third experiment we stretch the controllers to the limit by increasing the number of FTP flows to 400. The three plots are shown in Figure 6[1]. The PI AQM continues to exhibit acceptable performance, although it has become a little slower in it's response time. The two other AQMs, on the other hand, "hit the roof". This is a result of the fact that at such

[1] As another comparison point, we implement the stable Proportional controller derived in [8].

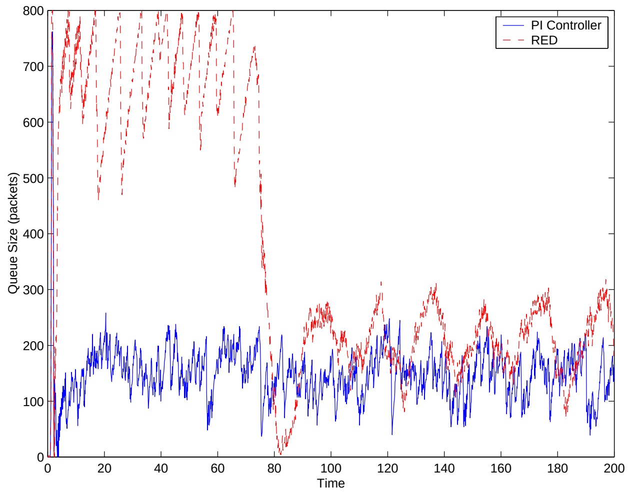

Figure 5: Experiment 2

high load factors, the loss probability has become so high that the steady state regulation error of those two controllers has pushed the operating queue length beyond the buffer size. This experiment illustrates the importance of integral control in an AQM system with a finite buffer.

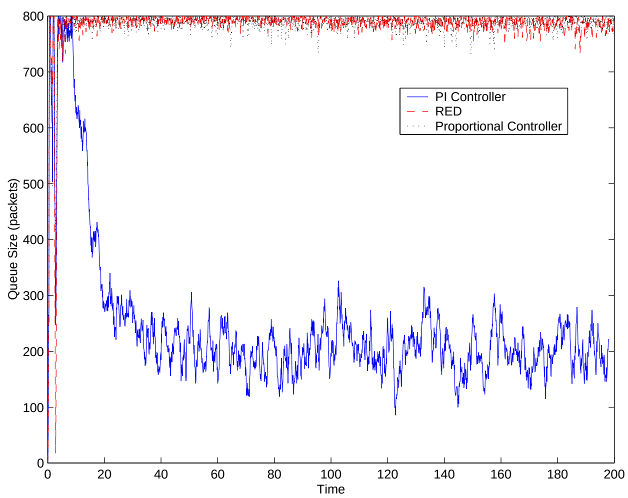

Figure 6: Experiment 3

4 The Adaptive Controller

The high-frequency gain of the open-loop transfer function (1) given by

$$\lim_{s\to\infty} P(s) = \frac{\frac{C^2}{2N}}{s^2} \doteq \frac{K_{hf}}{s^2}.$$

is a key measures of the system's performance. A lower high-frequency gain implies a lower system bandwidth. And lower bandwidth implies slower system in terms of rise and settling times. If K_{hf} belongs to a given range, then in order to insure 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, there exists an unavoidable limitation on the achievable performance with PI AQM. To see this, consider the case where the TCP load N can vary in the range $[N_{min}, N_{max}]$. Given that the fixed controller was designed using N_{max}, the plant's high-frequency gain is $\frac{N_{min}}{N_{max}}$ smaller. 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 to overcome such sluggishness.

One such controller, described below, is the Externally Excited Adaptive Loop (EEAL) algorithm specifically tailored to uncertainty in the high-frequency gain. The EEAL concept is a form of dithered feedback system (e.g., see [12]) sometimes referred to, in this context, as a self-oscillating adaptive system (SOAS). The SOAS and the externally excited adaptive system (EEAS) employ a relay element in the main feedback loop. In both forms, the system exhibits limit cycle whose amplitude can be used to modify the loop's gain. This was shown to make the control system insensitive to changes in the high-frequency gain of the plant. However, these forms suffer from serious limitations (e.g., see [9]-[12]). Several improvements have been presented, with the most recent termed *new EEAL* in [9] and [11]. This newest form of EEAL does not require a relay element and uses a separate identifier and loop gain changer. The result is a reduced sensitivity to sensor noise. However, it was found empirically that in order for the adaptation to work, all external inputs must be set to zero initially. This limitation, as well as the potentially bursty nature of TCP flows, introduced a difficulty when we applied it to our problem. The results reported here are based on several modifications to the algorithms in [9]-[11] tailored to our problem.

At this point, the simulations described below were done in `Simulink`. We compare the performance of the PI AQM. We model the number of TCP flows N using

$$N = 100 + \text{white noise} \in [50, 400]$$

where the white noise is a *Simulink* band-limited white noise with noise power of 10000 and Sample Time of 10. Queue levels under PI and EEAL AQMs are compared in Figure (7). The observed improvement is due to the adaptive nature of EEAL.

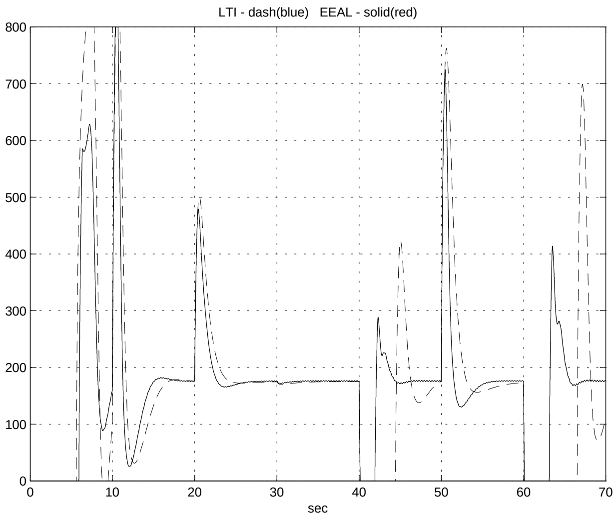

Figure 7: PI and EEAL AQMs response with stochastic load variation.

An additional test was run to compare the ability of an AQM to cope with a sudden flow burst causing queue saturation (this is done by increasing N at $t = 30$ sec from 100 to 400 flows). This step change results in the effective loop's bandwidth being cut by a factor of 4. While the gain of the PI controller remains fixed, that is, the PI loop is now 4 times more sluggish, the EEAL controller is able to increase its loop gain resulting in a faster response. The clear improvement in the queue level seen in Figure 8 occurs even with the adaptation reaching only 50% of its steady-state value.

5 Conclusions

In this paper we have proposed and designed two alternative AQMs to the RED AQM. The controllers in our AQMs were designed based on a new fluid model TCP developed in [5]. The first AQM, based on the classical PI controller, has many desirable properties: it is very simple to implement in real systems and shows good robustness. We implemented the PI AQM in `ns` and compared performance under various scenarios with RED AQM. The PI AQM exhibited superior performance. The second AQM, based on the adaptive EEAL algorithm, in addition to enjoying the PI AQM's advantage over RED AQM, also maintained consistent performance with varying number of TCP flows. The EEAL algorithm is more complex than the PI and owing to this complexity requires further investigation.

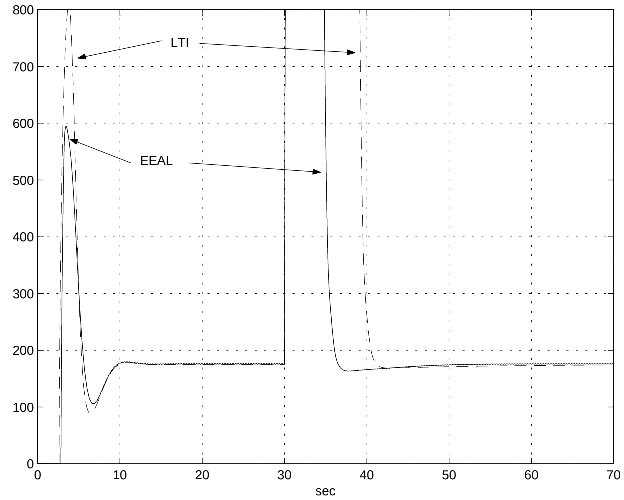

Figure 8: PI and EEAL AQMs responses to a flow burst at $t = 30$.

References

[1] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, 1997, Vol. 1(4), pp. 397-413.

[2] D. Lin and R. Morris, "Dynamics of Random Early Detection," *Proceedings of ACM/SIGCOMM*, 1997. pp. 127-137.

[3] W. Feng, D. Kandlur, D. Saha and K. Shin, *Blue: A New Class of Active Queue Management Algorithms*, Tech Report UM CSE-TR-387-99, Real-Time Computing Lab, EE Dept, UMich, 1999.

[4] Teunis J. Ott, T. V. Lakshman and L. H. Wong, "SRED: Stabilized RED," *Proceedings of Infocom*, 1999.

[5] C.V. Hollot, V. Misra, D. Towsley, W. Gong, "A Control Theoretic Analysis of RED," 2000; A version is currently available as UMass CMPSCI

Technical Report 00-41 (download from http://www-net.cs.umass.edu/papers/papers.html).

[6] `ns`, UCB/LBNL/VINT Network Simulator - ns (version 2), http://www.isi.edu/nsnam/ns/.

[7] 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," *Proceedings of ACM/SIGCOMM*, 2000, pp 151-160.

[8] C.V. Hollot, V. Misra, D. Towsley and W.B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," Submitted for review, ftp://gaia.cs.umass.edu/pub/Misra00-AQM-Controller.ps.gz, 2000.

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

[10] 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.

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

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