PD-RED: To Improve the Performance of RED

Jinsheng Sun, King-Tim Ko, *Member, IEEE*, Guanrong Chen, *Fellow, IEEE*, Sammy Chan, *Member, IEEE*, and Moshe Zukerman, *Senior Member, IEEE*

***Abstract*—In this letter, we propose a new active queue management (AQM) scheme to improve the performance of the well-known random early detection (RED) AQM. The new AQM is based on the proportional derivative (PD) control principle, and we call it PD-RED. In PD-RED we introduce minimal changes to RED. We demonstrate the improvement in performance of PD-RED over the recently introduced Adaptive RED AQM by simulations.**

***Index Terms*—Active queue management (AQM), Adaptive RED, congestion control, control theory, RED, TCP.**

I. INTRODUCTION

INTERNET congestion control has two parts: 1) the end-to-end protocol TCP and 2) the active queue management (AQM) scheme implemented in routers. AQM complements TCP by signaling congestion. This is done by discarding or marking packets during congestion. When congestion is detected by TCP, it will take actions to reduce the send rate. There are two types of AQM [18]: 1) rate based (e.g., [1]) which controls the flow rate at the congested link and 2) queue based (e.g., [5], [6]) which controls the queue at the congested link. The goals of queue based AQM are: 1) to stabilize the queue length at a given target thereby present predictable maximum queuing delay and 2) to minimize the occurrences of queue overflow and underflow, thus maximizing link utilization.

The random early detection (RED) is a very well-known queue-based AQM [6]. The key idea of RED is to keep the average buffer occupancy low. RED randomly discards an incoming packet with a probability proportional to the average buffer occupancy. So far, RED is the most prominent and widely studied AQM. However, it is very difficult to parameterize RED in order to obtain good performance under different congestion scenarios [10], [13]. RED attempts to stabilize the average buffer occupancy on a target value, but the current version of RED is not successful in achieving this goal because the average buffer occupancy strongly depends on the traffic loads and the parameter settings. In particular, RED does not perform well when the average buffer occupancy becomes larger than a certain threshold, resulting in decrease in throughput and increase in drop-rate.

There are many publications on RED, its variants and other AQM schemes (see, for example, [2], [4], [5], [7]–[9], [12], [15]–[17]). Out of the above publications, [2], [9], [16], and [17] have used control theory fundamentals to analyze and develop new AQMs. However, none of them used fundamental means to enhance RED. This letter proposes a new Adaptive RED scheme based on the proportional derivative (PD) control principle, called PD-RED. This scheme is based on control theory and adapts the maximal drop rate parameter of RED called max_p to stabilize the queue length.

II. PD-RED ALGORITHM

An important objective of (queue based) AQM is to stabilize the queue length at a given target. The benefits of a stabilized queue length in a network are high resources utilization, predictable maximum delays, and traffic-load-independent network performance in terms of traffic intensity and number of connections. To this end, Floyd *et al.* proposed the Adaptive RED scheme [5] (an enhancement of [4]). In their scheme, the parameter max_p is adapted using an additive-increase multiplicative-decrease policy to keep the average queue length within a target range. Although Adaptive RED can stabilize the queue length at a given target, its performance can still be improved if we adapt the drop probability max_p in a more methodological manner.

We can consider AQM as a typical control system. Let time be divided into consecutive sampling intervals each of size δt. Let Q_T be the target queue occupancy and $q_{\text{avg}}(i)$ the average queue length during sampling interval i. Our goal is to adapt max_p, every sampling time interval, so that the magnitude of the error signal

$$e(i) = q_{\text{avg}}(i) - Q_T \tag{1}$$

for sampling interval i, is kept as small as possible.

Despite significant control theory advancements, most industrial processes use *proportional-integral-derivative* (PID) *controllers*. This demonstrates the great potential of the simple PID control strategy. Therefore, we propose to add a PD controller to RED, so our proposed PD-RED AQM solution is composed of two parts: 1) a new PD controller and 2) the original RED AQM. The PD controller can be described as follows:

$$max_p(i) = max_p\,(i-1) + k_p e(i) + k_d(e(i) - e(i-1)) \tag{2}$$

where k_p is the proportional gain and k_d is the derivative gain.

Manuscript received January 23, 2003. The associate editor coordinating the review of this letter and approving it for publication was Dr. J. Kim. This work described was supported in part by a grant from the Research Grants Council of the Hong Kong SAR, China, under Project CityU 1031/01E.

J. Sun is with the Department of Automation, Nanjing University of Science and Technology, Nanjing 210094, China.

K.-T. Ko, G. Chen, and S. Chan are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China.

M. Zukerman is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, SAR, China, on leave from the ARC Special Research Centre for Ultra-Broadband Information Networks, EEE Department, The University of Melbourne, Australia (e-mail: m.zukerman@ee.mu.oz.au).

Digital Object Identifier 10.1109/LCOMM.2003.815653

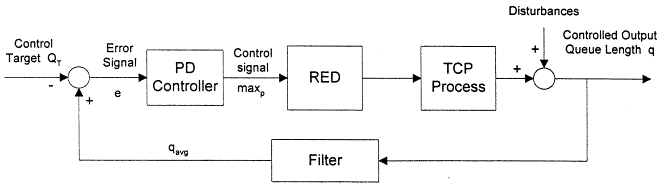


Fig. 1. Block diagram of TCP congestion control mechanism as a closed-loop feedback control system.

To maintain $0 \leq max_p(i) \leq 1$, we set $max_p(i) = 0$ if $max_p(i) < 0$, and $max_p(i) = 1$ if $max_p(i) > 1$, for all i.

The rule industry goes by is that routers should be able to store data equal to the bandwidth-delay product [3], [11]. Therefore, we can assume that buffers grow linearly in transmission rate. Since queue length fluctuations grow linearly with the bandwidth-delay product, they also grow linearly with the buffer size denoted B. Therefore, we use the normalized error signal $(e(i)/B)$ instead of the error signal. Replacing $e(i)$ with $e(i)/B$ in (2) results in the following control equation:

$$max_p(i) = max_p(i-1) + k_p \frac{e(i)}{B} + k_d \frac{(e(i) - e(i-1))}{B}. \tag{3}$$

A simplified block diagram that represents the various components of the TCP/AQM closed-loop feedback control system including the two parts of PD-RED (the PD Controller and the original RED) is shown in Fig. 1.

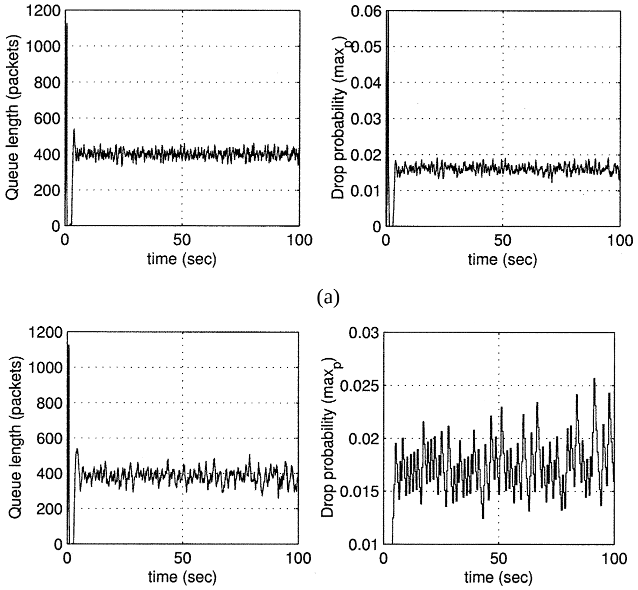
The computations of $max_p(i)$ for time i (the ith sampling interval) can be summarized as follows:

1) sample average queue length $q_{\text{avg}}(i)$;
2) compute current error signal $e(i)$ by (1);
3) compute current drop probability $max_p(i)$ by (3);
4) use $max_p(i)$ in RED as the drop probability until time $i+1$, when a new max_p is to be computed again;
5) store $e(i)$ and $max_p(i)$ to be used at time $i+1$.

III. PERFORMANCE COMPARISONS

In this section, we compare PD-RED and Adaptive RED by simulations performed using the *NS* simulator [14]. Comparisons with other RED variants will be reported elsewhere. The network topology used in the simulation is the same one used in [5]. It is a simple dumbbell topology based on a single common bottleneck link of 45 Mb/s capacity with many identical, long-lived and saturated TCP/Reno flows. In other words, the TCP connections are modeled as greedy FTP connections, that always have data to send as long as their congestion windows permit. The receiver's advertised window size is set sufficiently large so that the TCP connections are not constrained at the destination. The ack-every-packet strategy is used at the TCP receivers.

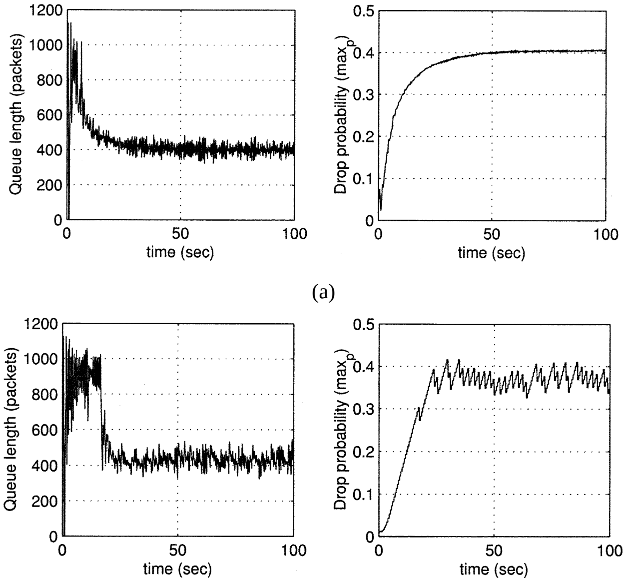
For the two AQM schemes tested, we maintain the same test conditions: the same topology (as described above), the same saturated traffic and the same TCP parameters. The parameters used are as follows: the round-trip propagation delay is 100 ms, packet size is 500 bytes, the total buffer size is set at 1125 packets (twice the bandwidth-delay product of the network). The basic parameters of RED (see notation in [6]) are set at $min_{\text{th}} = 15$, $max_{\text{th}} = 785$, $max_p = 0.01$ and $w_q = 0.002$. For Adaptive RED, the parameters are set as the same in [5]: $\alpha = 0.01$, $\beta = 0.9$, $intervaltime = 0.5$ s. For PD-RED, we set $\delta t = 0.01$ s, $k_p = 0.001$ and $k_d = 0.05$. For both AQMs, the target is set at $target = (max_{\text{th}} + min_{\text{th}})/2 = 400$.



(a)

(b)

Fig. 2. Queue length and drop probability for 100 connections. (a) PD-RED. (b) Adaptive RED.

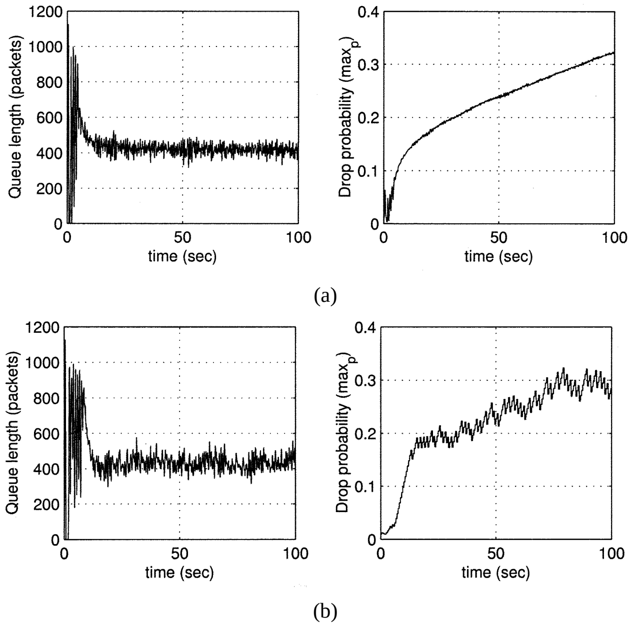


(a)

(b)

Fig. 3. Queue length and drop probability for 2000 connections. (a) PD-RED. (b) Adaptive RED.

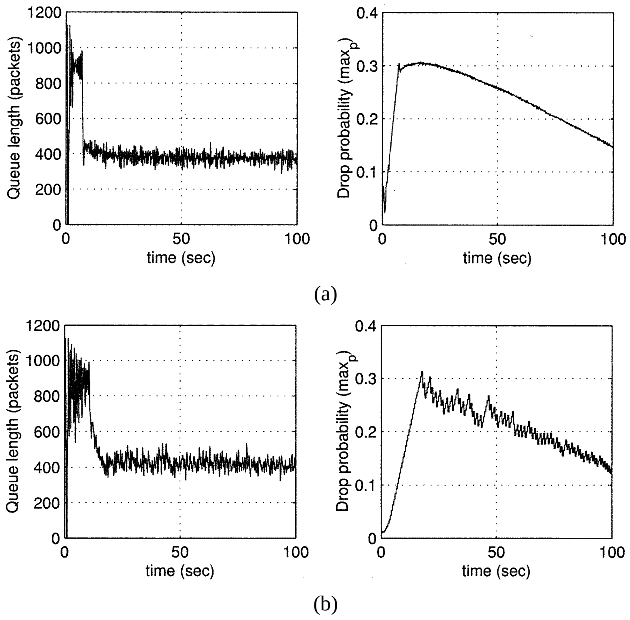
First, we test the performance for a constant number of TCP connections. Fig. 2 presents the queue length and the drop probability for 100 connections. We can see that the fluctuation amplitude of PD-RED queue length is smaller and the variance of the drop probability is much smaller than those of Adaptive RED. To compare the steady-state performance, we calculate the average and the standard deviation of queue length for second half (50 s). We found that the mean queue length for PD-RED is 399.9 (just on the 400 target) versus 379.1 for Adaptive RED. The standard deviation of the queue length is 22.2 for PD-RED versus 46.8 for Adaptive RED. Fig. 3 presents the queue length and the drop probability for 2000 connections. We can see that PD-RED reacts faster than Adaptive RED, the



(a)

(b)

Fig. 4. Queue length and drop probability for variable number of connections starting at 500 (a) PD-RED (b) Adaptive RED.



(a)

(b)

Fig. 5. Queue length and drop probability for variable number of connections starting at 1500 (a) PD-RED (b) Adaptive RED.

queue length fluctuation amplitude of PD-RED is smaller than that of Adaptive RED, and the drop probability of Adaptive RED fluctuates more. The mean queue length for PD-RED is 401.6 (just above the 400 target) versus 435.6 for Adaptive RED. The standard deviation of the queue length is 27.9 for PD-RED versus 52.9 for Adaptive RED.

We now compare the performance of PD-RED with Adaptive RED in cases where the number of TCP connections varies. We performed two simulations. In the first, the initial number of connections is set to 500 and, in addition, 1000 connections have their start-time uniformly distributed over a period of 100 s. In the second simulation, the initial number of connections is set to 1500 and then 1000 connections have their stop-time uniformly distributed over a period of 100 s.

Figs. 4 and 5 present the queue length and the drop probability for the 1st and 2nd simulations, respectively. In both cases, we can see that PD-RED reacts faster than Adaptive RED, its queue length fluctuation amplitude is smaller and its drop probability function is much smoother than that of Adaptive RED. Specifically, in the first case, the mean queue length is 419.0 for PD-RED versus 431.2 for Adaptive RED, and the standard deviation of queue length is 33.7 for PD-RED versus 51.2 for Adaptive RED. In the second case, the mean queue length is 374.9 for PD-RED versus 417.7 for Adaptive RED. The standard deviation of queue length is 36.24 for PD-RED versus 36.75 for Adaptive RED. We can see that while in the first case PD-RED performs clearly better than Adaptive RED. In the second case, Adaptive RED is slightly closer to the 400 target.

IV. CONCLUSIONS

We have proposed a new variant RED scheme—called PD-RED, which is based on control theory fundamentals. We have demonstrated by simulations that PD-RED performs better than Adaptive RED in most cases.

REFERENCES

[1] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, May/June 2001.
[2] J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Comput. Networks*, vol. 36, no. 2–3, pp. 203–235, 2001.
[3] Cisco. Cisco 12000 Series Routers, San Jose, CA. [Online]. Available: http://www.cisco.com/en/US/products/hw/routers/ps167/index.html
[4] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A self-configuring RED gateway," in *Proc. INFOCOM 1999*, vol. 3, Mar. 1999, pp. 1320–1328.
[5] S. Floyd, R. Gummadi, S. Shenker, and ICSI. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management, Berkeley, CA. [Online]http://www.icir.org/floyd/red.html
[6] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397–413, Aug. 1993.
[7] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proc. IEEE INFOCOM 2000*, vol. 3, Tel-Aviv, Israel, Mar. 2000, pp. 1435–1444.
[8] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "A control theoretic analysis of RED," in *Proc. IEEE INFOCOM 2001*, vol. 3, Anchorage, AK, Apr. 2001, pp. 1510–1519.
[9] ——, "On designing improved controllers for AQM routers supporting TCP flows," in *Proc. IEEE INFOCOM 2001*, vol. 3, Anchorage, AK, Apr. 2001, pp. 1726–1734.
[10] G. Iannaccone, M. May, and C. Diot, "Aggregate traffic performance with active queue management and drop from tail," *ACM SIGCOMM Computer Communication Rev.*, vol. 31, no. 3, pp. 4–13, July 2001.
[11] Juniper T-series Routers, Sunnyvale, CA.. [Online] Available: https://www.juniper.net/products/ip_infrastructure/t_series/100051.html
[12] D. Lin and R. Morris, "Dynamics of random early detection," in *Proc. ACM SIGCOMM 1997*, 1997, pp. 127–137.
[13] M. May, J. Bolot, C. Diot, and B. Lyles, "Reasons not to deploy RED," in *Proc. 7th. Int. Workshop on Quality of Service (IWQoS'99)*, London, U.K., Mar. 1999, pp. 260–262.
[14] USC/ISI, Los Angeles, CA. The NS simulator and the documentation. [Online] Available: http://www.isi.edu/nsnam/ns/
[15] T. J. Ott, T. V. Lakshman, and L. Wong, "SRED: Stabilized RED," in *Proc. IEEE INFOCOM*, vol. 3, New York, Mar. 1999, pp. 1346–1355.
[16] F. Ren, C. Lin, X. Ying, X. Shan, and F. Wang, "A robust active queue management algorithm based on sliding mode variable structure control," in *Proc. IEEE INFOCOM*, vol. 1, New York, June 2002, pp. 13–20.
[17] F. Ren, Y. Ren, and X. Shan, "Design of a fuzzy controller for active queue management," *Comput. Commun.*, vol. 25, no. 9, pp. 874–883, June 2002.
[18] B. Wydrowski and M. Zukerman, "QoS in best-effort networks," *IEEE Commun. Mag.*, vol. 40, pp. 44–49, Dec. 2002.