On Designing a Novel PI Controller for AQM Routers Supporting TCP Flows

Nai-xue Xiong[1], Yan-xiang He[1], Yan Yang[2],
Bin Xiao[3], and Xiaohua Jia[4]

[1] The State Key Lab of Software Engineering,
Computer School of Wuhan University, Wuhan Hubei, 430079, PR. China
N.Xiong@mail.ccnu.edu.cn, yxhe@whu.edu.cn
[2] Computer Science Department of Central China Normal University,
Wuhan 430079, PR. China
Y.Yang@mail.ccnu.edu.cn
[3] Department of Computing, Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
bxiao@utdallas.edu
[4] SMIEEE, Member of IEEE Communication Society,
Department of Computer Science,
City University of Hong Kong, Kowloon, Hong Kong
csjia@cityu.edu.hk

Abstract. Active Queue Management (AQM) is an effective method to enhance congestion control, and to achieve tradeoff between link utilization and delay. The de facto standard, Random Early Detection (RED), and most of its variants use queue length as congestion indicator to trigger packet dropping. As an extension of RED, a novel AQM algorithm, called NPI-RED, is proposed in this paper. The NPI-RED is based on a novel proportional and integral controller, which not only considers the average queue length at the current time slot, but also takes into consideration the past average queue lengths within a round trip time. We provide a guideline for the selection of the feedback gains for TCP/RED system to stabilize the dynamics, make the queue length converge at a certain target and improve the network performance. We present the condition of asymptotic stability for the model in terms of the average queue length, by using a method, in which we construct a Routh table associated with the characteristic polynomial. Based on the stability condition and control gains selection method, the extensive simulation results by ns2 demonstrate that the NPI-RED algorithm outperforms than the existed AQM schemes in robustness, drop probability and stability.

1 Introduction

Internet congestion occurs when the aggregated demand for a resource (e.g., link bandwidth) exceeds the available capacity of the resource. Resulting effects from such congestion include long delays in data delivery, waster resources due to lost or

Y. Zhang et al. (Eds.): APWeb 2005, LNCS 3399, pp. 991–1002, 2005.

dropped packets, and even possible congestion collapse [1]. Therefore it is very necessary to avoid or control network congestion. Internet congestion control has two parts: 1) the end-to-end protocol TCP and 2) the active queue management (AQM) scheme implemented in routers [2]. AQM can maintain smaller queuing delay and higher throughput by purposefully dropping packets at intermediate nodes.

Several AQM Schemes have been studied in recent literatures to provide early congestion notification to users, e.g., random early detection (RED) [3] and its variant [4], such as Proportional Integral (PI) controller [5], REM [1, 6], BLUE [7], adaptive virtual queue (AVQ) algorithm [8] and PD-RED controller [2]. These schemes can be classified: 1) rate based which controls the flow rate at the congested link and 2) queue based which controls the queue at the congested link. RED [3] is the most prominent and widely studied AQM scheme, which is implemented in routers for congestion control of the Internet. Dynamic-RED (DRED) attempts to stabilize router queue occupancy at a level independent of the active connections by using EWMA as an integral control (I-control) [9]. In [5], the proportional-integral (PI)-controller has been proposed to improve responsiveness of the TCP/AQM dynamics and to stabilize the router queue length around the target. Based on a Proportional-Integral-Derivative (PID) model, an adaptive control mechanism is proposed to improve the system stability and performance under changing network conditions [10]. In these approaches, feedback control theory is used to describe and analyze the TCP/AQM dynamics. However, the current version of RED does not succeed in the goal of stabilizing the queue length. The main reason is probably that the present RED does not use any exact mathematical model to characterize the complex TCP congestion control process; it is thus difficult to provide any systematic and robust law to configure RED's parameters. Unfortunately, its control gain selection is based on empirical investigation and simulation analysis. As can be expected, this method is often ad hoc in nature, and may only be useful for certain class of processes or under certain conditions. A theoretic guideline to choose proper control gain to optimize network performance is still required.

In this paper, a novel proportional integral (NPI) feedback control scheme is proposed for TCP/RED dynamic model developed in [11, 12]. By using control theory and regulating the router queue length to approach the expectative value, the proposed AQM algorithm can decrease the responsive time and improve the stability and robustness of TCP/AQM congestion control. Based on the time-delay control theory, we investigate its asymptotic stability and provide a guideline for the selection of the feedback gains (the proportional and integral parameters) for TCP/RED system, which can make the queue length converge at the expected target, stabilize the RED and improve the network performance.

The remainder of the paper is organized as follows. In Section 2, we present a novel algorithm called NPI-RED and give a theoretic law of choosing the proportional and integral parameters to achieve the system stability. The simulation results demonstrate that the better network performance of NPI-RED can be achieved compared with other AQM schemes in different network conditions by ns2 in Section 3. Finally, we conclude our work and give the further work in section 4.

2 The NPI-RED Algorithm

2.1 Algorithm Description

There are three parameters, i.e., $\{q_{ref}, K_p, K_I\}$ that need to be set in RED. They are specified in the table 1.

We consider the following dropping formula as the feedback controller:

$$\delta p(t) = \frac{K_p (avgq(t) - q_{ref})}{B} - \frac{K_I}{B} \int_{(t-1)-R}^{t-1} (avgq(v) - q_{ref}) dv, \tag{1}$$

where $avgq(t)$ denotes the average queue length at time t. We use the average queue length in calculating drop probability instead of the current queue length in order to avoid the sudden change of queue length causing by some short flows or non-TCP flows. From the view of control theory, it is very necessary to ensure the network system stability for an effective system. If the control gains are selected in the stability areas, it can enable the system stability and therefore the network may have better performance, such as less loss ratio, less queue delay and higher network throughput. Then we make the analysis on the system stability and give a theoretic guideline to choose proper control gains to optimize network performance.

Table 1. Parameters of RED

Parameter	Description
q_{ref}	Desired queue size (packets);
K_p	The proportional control gain;
K_I	The integral control gain;
B	The buffer size of the congested router;
p	Probability of packet drop

2.2 Stability Analysis and Control Gain Selection

The stability of the congested buffer occupancy is significantly important because the large oscillation in the buffer can cause a large of data drop, the lower link utilization and the lower system throughput. It influence seriously on the Quality of Service (QoS) of the network. Therefore, in order to meet QoS requirement (e.g., acceptable delay), it is important to consider the stability of the bottleneck link and reduce the steady-state queue length at the routers, and this is the objective of queue management. In this section, we make analysis on the stability based on the dynamic model of TCP and the control theoretic approach, and give the design method in detail. We use the simplified dynamic model of TCP behavior [12] as shown in the following equations:

$$\begin{cases} \delta\dot{W}(t) = -\dfrac{2N}{R_0^2 C}\delta W(t) - \dfrac{R_0 C^2}{2N^2}\delta p(t-R_0) \\ \delta\dot{q}(t) = \dfrac{N}{R_0}\delta W(t) - \dfrac{1}{R_0}\delta q(t) \end{cases} \tag{2}$$

In system (2), we denote $\delta W(t) = W(t) - W_0$, $\delta q(t) = q(t) - q_0$, $\delta p(t) = p(t) - p_0$, where (W_0, q_0, p_0) is the equilibrium point of the system. Here, $\dot{W}(t)$ and $\dot{q}(t)$ denote the time-derivative of $W(t)$ and $q(t)$ respectively, and we denote the parameters of equations (2) in Table 2.

Table 2. Parameters of model (2)

Parameter	Description
W	Expected TCP window size (packets)
q	Current queue length (packets)
R_0	Round-trip time (second)
C	Link capacity (packets/second)
N	Load factor (number of TCP connections)

In [7], Exponentially Weighted Moving Average (EWMA) queue size is used, i.e., the average queue size *avgq*(*n*) as a measure of congestion of the network. Specifically, with RED, a link maintains the following equation

$$avgq(n) = (1 - w_q)avgq(n-1) + w_q q(n) \tag{3}$$

where *avgq*(*n*) denotes the average queue length at the n^{th} interval, $q(n)$ is the instantaneous queue size at the n^{th} interval and w_q is a weight parameter, $0 \le w_q \le 1$. Let the sampling interval equal to 1, and then we can reach the continuous-time form of equation (3) as follows:

$$q(t) = \frac{1}{w_q}avgq(t) + (1 - \frac{1}{w_q})avgq(t-1) \tag{4}$$

By differentiating both sides of equation (4), we have

$$\dot{q}(t) = \frac{1}{w_q}avg\dot{q}(t) + (1 - \frac{1}{w_q})av\dot{g}q(t-1) \tag{5}$$

In this paper our objective is to develop the active queue management to improve the stability of the bottleneck queue in the network. A linear dynamic TCP model in document [12] is introduced in this paper.

By substituting (4) and (5) into (2), we have

$$\begin{cases} \delta\dot{W}(t) = -\dfrac{2N}{R_0^2 C}\delta W(t) - \dfrac{R_0 C^2}{2N^2}\delta p(t - R_0) \\ \dfrac{1}{w_q} avg\dot{q}(t) + \left(1 - \dfrac{1}{w_q}\right) avg\dot{q}(t-1) + \dfrac{1}{R_0}\left(\dfrac{1}{w_q} avgq(t) + \left(1 - \dfrac{1}{w_q}\right) avgq(t-1)\right) \\ \qquad = \dfrac{N}{R_0}\delta W(t) + \dfrac{1}{R_0} q_0 \end{cases} \quad (6)$$

The dynamics (2) and (6) are illustrated in the following block diagram:

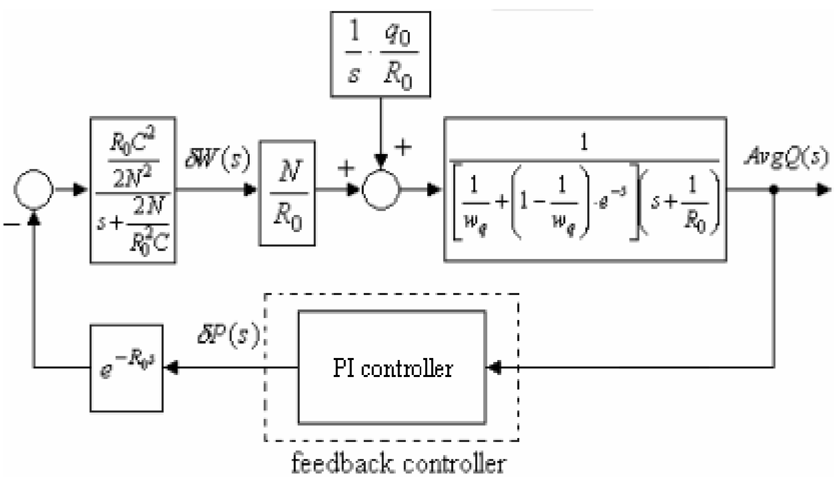


Fig. 1. Block-diagram of the dynamics (2) and (6)

By performing Laplace Transform of (5) and (6), we can derive

$$\begin{cases} \left(s + \dfrac{2N}{R_0^2 C}\right)\delta W(s) = -\dfrac{R_0 C^2}{2N^2}\cdot e^{-R_0 s}\cdot \delta P(s) \\ \left[\dfrac{1}{w_q} + \left(1 - \dfrac{1}{w_q}\right)\cdot e^{-s}\right]\left(s + \dfrac{1}{R_0}\right) AvgQ(s) = \dfrac{N}{R_0}\delta W(s) + \dfrac{1}{s}\cdot\dfrac{q_0}{R_0} \\ \delta P(s) = \dfrac{K_P}{B}\left(AvgQ(s) - \dfrac{q_{ref}}{s}\right) - \dfrac{K_I}{Bs}\left(e^{-s} - e^{-(R_0+1)s}\right) AvgQ(s) \end{cases} \quad (7)$$

where $\delta W(s)$, $AvgQ(s)$, $\delta P(s)$ denote the Laplace Transform of $\delta W(t)$, *avgq(t)*, $\delta p(t)$ respectively. For similarity, we assume that $K_P > 0$, $K_I > 0$. Finally, we can get the following polynomial equation

$$\Delta(s)AvgQ(s) = -\frac{R_0 C^2 e^{-R_0 s}}{2}\frac{K_p q_{ref}}{Bs} - (s + \frac{2N}{R_0{}^2 C})\frac{Nq(0)}{s},$$

where $\Delta(s)$ is the characteristic equation is

$$\Delta(s) = -(s + \frac{2N}{R_0{}^2 C})[\frac{1}{w_q} + (1 - \frac{1}{w_q})e^{-s}](NR_0 s + R_0 s + 1) + \frac{R_0 C^2}{2N}e^{-R_0 s}[\frac{K_P}{B} - \frac{K_I}{Bs}(e^{-s} - e^{-(R_0+1)s})].$$

Then we suppose $e^{-R_0 s} = 1 - R_0 s + \frac{1}{2}R_0 s^2$ and the $\Delta(s)$ is the characteristic equation

$$\Delta(s) = s^4[\frac{1}{2}R_0{}^3(1 - \frac{1}{w_q})] + s^3[-R_0{}^2(1 - \frac{1}{w_q}) + \frac{1}{2}(1 - \frac{1}{w_q})(R_0{}^2 + \frac{2NR_0}{C}) + \frac{R_0{}^5 C^2 K_I}{8NB} + \frac{R_0{}^4 C^2 K_I}{4NB}] + s^2[R_0 - R_0(1 - \frac{1}{w_q})(1 + \frac{2N}{R_0 C}) + \frac{N}{C}(1 - \frac{1}{w_q}) + \frac{R_0{}^3 C^2 K_P}{4NB} - \frac{R_0{}^4 C^2 K_I}{2NB} - \frac{R_0{}^3 C^2 K_I}{2NB}] + s^1[1 + \frac{2N}{R_0 C w_q} - \frac{R_0{}^2 C^2 K_P}{2NB} + \frac{3R_0{}^3 C^2 K_P}{4NB} + \frac{R_0{}^2 C^2 K_I}{2NB}] + s^0(\frac{R_0 C^2 K_P}{2NB} + \frac{2N}{R_0{}^2 C} - \frac{R_0{}^2 C^2 K_I}{2NB}).$$

Suppose the coefficients in the above equation of s^4, s^3, s^2, s^1 and s^0 are respectively a_4, a_3, a_2, a_1 and a_0, i.e.,

$$a_4 = \frac{1}{2}R_0{}^3(1 - \frac{1}{w_q}),$$

$$a_3 = -R_0{}^2(1 - \frac{1}{w_q}) + \frac{1}{2}(1 - \frac{1}{w_q})(R_0{}^2 + \frac{2NR_0}{C}) + \frac{R_0{}^5 C^2 K_I}{8NB} + \frac{R_0{}^4 C^2 K_I}{4NB},$$

$$a_2 = R_0 - R_0(1 - \frac{1}{w_q})(1 + \frac{2N}{R_0 C}) + \frac{N}{C}(1 - \frac{1}{w_q}) + \frac{R_0{}^3 C^2 K_P}{4NB} - \frac{2R_0{}^4 C^2 K_I}{4NB} - \frac{R_0{}^3 C^2 K_I}{2NB},$$

$$a_1 = 1 + \frac{2N}{R_0 C w_q} - \frac{R_0^{\ 2} C^2 K_P}{2NB} + \frac{3R_0^{\ 3} C^2 K_I}{4NB} + \frac{R_0^{\ 2} C^2 K_I}{2NB} \text{ and}$$

$$a_0 = \frac{R_0 C^2 K_P}{2NB} + \frac{2N}{R_0^{\ 2} C} - \frac{R_0^{\ 2} C^2 K_I}{2NB}.$$

Based on the Routh-Hurwitz stability test [13], we get the following Routh Table (in Table 3), where $\gamma_{31} = \dfrac{a_3 a_2 - a_4 a_1}{a_3}$ and $\gamma_{41} = \dfrac{r_{31} a_1 - a_3 a_0}{r_{31}}$.

The whole network system is stable if and only if the valves of the second column are all greater than zero, i.e., $a_4 > 0$, $a_3 > 0$, $\gamma_{31} > 0$, $a_0 > 0$ and $\gamma_{41} > 0$ (i.e., $f(K_P) = a_1 a_2 a_3 - a_1^2 a_4 - a_0 a_3^2 < 0$). We set K_{P1} and K_{P2} are the roots of the function $f(K_P) = 0$ and $K_{P1} < K_{P2}$.Therefore, the range of control gains K_P and K_I in the case of the system stability are as follows:

Table 3. The Routh table

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2	γ_{31}	a_0	0
s^1	γ_{41}	0	0
s^0	a_0	0	0

$$0 < K_I < \frac{(w_q - 1)(0.5CR_0 - N)}{w_q R_0^{\ 3} C^3 (R_0 + 2)}, \quad K_P < R_0 K_I - \frac{4N^2 B}{R_0^{\ 3} C^3}, \quad \frac{(K_{P2} + K_{P1})}{2} < K_P < K_{P2},$$

and

$$K_P < \frac{0.5R^3 (\frac{1}{w_q} - 1)(1 + \frac{2N}{R_0 C w_q} + \frac{R_0^{\ 2} C^2 K_I}{2NB} + \frac{3R_0^{\ 3} C^2 K_I}{4NB})}{\frac{R_0^{\ 4} C}{4B} + \frac{R_0^{\ 5} C^2}{8NB} - \frac{R_0^{\ 4} C}{4Bw_q} - \frac{R_0^{\ 5} C^2}{8NBw_q} + \frac{R_0^{\ 7} C^4 K_I}{16NB^2} + \frac{R_0^{\ 8} C^4 K_I}{32NB^2}} + (R_0 + \frac{N(1 - w_q)}{Cw_q}$$

$$+ R_0 (1 + \frac{2N}{R_0 C})(1 - \frac{1}{w_q}) + \frac{R_0^{\ 3} C^2 K_I (1 - R_0)}{NB})[R_0^{\ 2} (\frac{1}{w_q} - 1) + (\frac{NR_0}{C} + \frac{R_0}{2})(1 - \frac{1}{w_q})$$

$$+ \frac{C^2 R_0^{\ 4} K_I (1 + \frac{R_0}{2})}{4NB}] / (\frac{R_0^{\ 4} C}{4B} + \frac{R_0^{\ 5} C^2}{8NB} - \frac{R_0^{\ 4} C}{4Bw_q} - \frac{R_0^{\ 5} C^2}{8NBw_q} + \frac{R_0^{\ 7} C^4 K_I}{16NB^2} + \frac{R_0^{\ 8} C^4 K_I}{32NB^2}).\text{sss}$$

2.3 The Specific Algorithm of NPI-RED

Based on the above stability analysis, we can select the proper control gains that can ensure the system stability and therefore improve the network performance. The computations of *p(n)* for time *n* (the n^{th} sampling interval) can be summarized in Fig. 2.

1) Sample the average queue length *avgq*(*n*);
2) Compute current error signal by the equation $e(n) = avgq(n) - q_{ref}$;
3) Estimate the number of current active TCP connections *N* using the technique of flow monitoring and accounting [14], and measure the round trip time *R*; then compute the range of K_p and K_I based on the above stability condition.
4) Compute current drop probability by (3);
5) Use the equation $p(n) = \frac{K_P \cdot e(n)}{B} + K_I \sum_{i=n-R-1}^{n-1} e(i)$ in RED as the drop probability until time (*n* + 1) when a new *p* is to be computed again;
6) Store the *p*(*n*) to be used at time (*n* + 1).

Fig. 2. The NPI-RED algorithm

3 Performance Evaluation

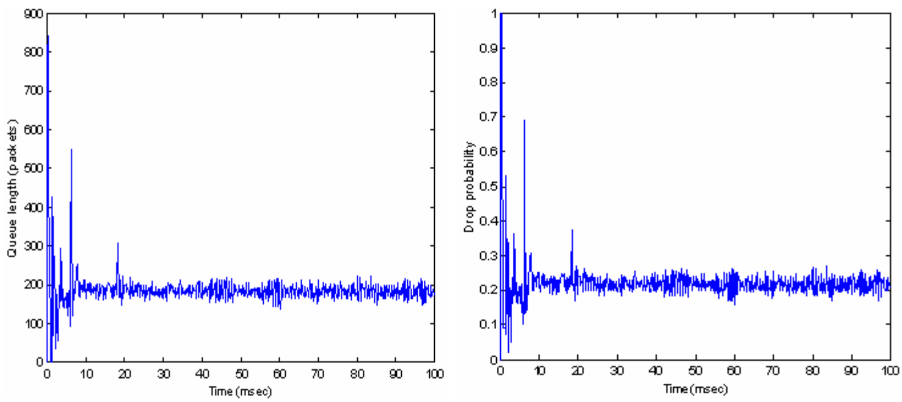
In this section, we evaluate the performance of the proposed NPI-RED algorithm by a number of simulations performed using ns2 [15]. The performance of NPI-RED is compared with RED [3] and other RED variants such as PI-RED [5], PD-RED [2] and adaptive RED [18]. The network topology used in the simulation is the same one used in [16-17]. 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 these 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, the mean packet size is 500 bytes, and the buffer size is set to be 1125 (twice the bandwidth-delay product of the network). The basic parameters of RED (see notation in [6]) are set at min_{th}=15, max_{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*=0.5s. For PD-RED, the parameters are set as same as [10], i.e., $\delta t = 0.01$, k_p = 0.001 and k_d = 0.05. For NPI-RED, we set $K_P = 12$ and $K_I = 0.01$.

In this simulation, we focus on the following key performance metrics: goodput (excluding packet retransmissions), average queue length, average absolute queue deviation, and packet loss ratio. The average queue length is defined as the arithmetic mean value of instantaneous queue length. The average queue deviation is defined as the absolute deviation between instantaneous queue length and its mean value.

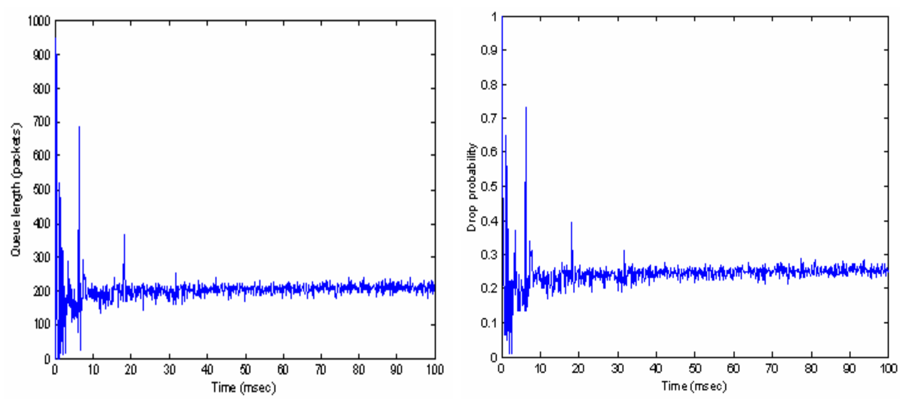
3.1 Simulation 1: Stability Under Extreme Conditions

In this experiment, all TCP flows are persistent, and the stability of the AQM schemes are investigated under two extreme cases: 1) light congestion with a small number of TCP flows *N* (*N=100*), 2) heavy congestion with a large N (*N=2000*). We set the queue target at 200. Other parameters are the same as those in the second paragraph of the performance evaluation part.



a) Queue length (b) Drop probability

Fig. 3. Light congestion (*N* = 100)



(a) Queue length (b) Drop probability

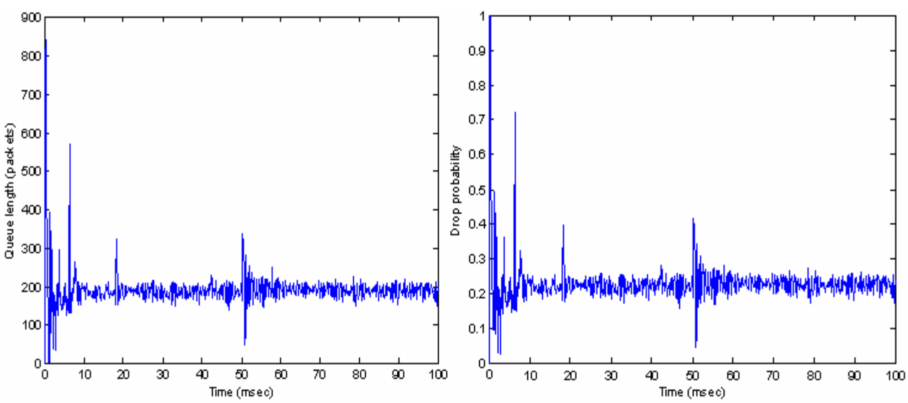
Fig. 4. Heavy congestion (*N* = 2000)

Fig. 3 and Fig. 4 demonstrate the dynamic change of the average queue length and the drop probability of the NPI-RED algorithm under the light congestion (*N* = *100*) and the heavy congestion (*N* = *2000*). It can be seen that, although the queue length and the drop probability fluctuate at first, they can be stable quickly, and be near to the queue target 200 and 0.22 respectively. Both the fluctuation amplitude of NPI-RED queue length and the variance of the drop probability are small. In summary, NPI-RED shows better stability and quick response under either light congestion or heavy congestion.

3.2 Simulation 2: Response Under the Variable Number of Connections

In this section, the simulation is performed with the variable number of TCP connections. In the first, the initial umber of connections is set to 2000 and, in addition, 200 TCP connections join in the link at 50.1 ms. Other parameters are the same as those in simulation 1.

Fig. 5 shows the queue length and the drop probability for the variable number of connections. We can find that in the first half queue length can be approximately stabilized at the queue target 200, and the drop probability can also be approximately stable at 0.22. At 50.1 ms, the queue length and the drop probability fluctuate with 200 TCP connections joining in the link, and after that they can be quickly stable. From these figures we can find that NPI-RED achieves a short response time, good stability and good robustness.



(a) Queue length (b) Drop probability

Fig. 5. Queue length and drop probability for variable number of connections starting at 2000

4 Conclusion and Future Work

In this paper, a novel AQM scheme called NPI-RED is proposed to improve the performance of RED. We have explored the queue length stability of NPI-RED, and have provided guidelines based on theory and simulations for control gain optimization.

Based on the stability condition and control gains selection method, the extensive simulation results by ns2 demonstrate that the NPI-RED scheme outperforms than recently proposed AQM algorithms in robustness, drop probability and stability.

Future work would cover the extension of the proposed approach from the model of a single bottleneck link only with TCP flows to the case of the multiple bottleneck links and to the case where the TCP and non-TCP traffic (e.g., UDP flows) share a single queue. The performance under short flows, and burst traffic loads will also be investigated. In addition, issues such as fairness and protection against non-responsive flows will be addressed, and all other relevant problems that the current RED is facing.

References

1. S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin: REM: active queue management, IEEE Networking, Vol. 15, pp. 48-53, May/June, 2001.
2. Jinsheng Sun, King-Tim Ko, Guanrong Chen, Sammy Chan, and Moshe Zukerman: PD-RED: to improve the performance of RED, IEEE Communications letters, vol. 7, no. 8, August 2003.
3. S. Floyd and V. Jacobson: Random early detection gateways for congestion avoidance, IEEE/ACM Trans. On Networking, Vol. 1, pp. 397-413, Aug. 1993.
4. E. C. Park, H. Lin, K. J. Park, and C. H. Choi: Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks, Computer Networks, vol. 44, no. 1, pp. 17-41, 2004.
5. C. V. Hollot, V. Misra, D. Towsley and W. B. Gong: Analysis and design of controllers for AQM routers supporting TCP flows, IEEE Transactions on Automatic Control, Vol. 47, pp. 945-959, Jun. 2002.
6. C. N. Long, J. Wu and X. P. Guan: Local stability of REM algorithm with time-varying delays, IEEE Communications Letters, Vol. 7, pp.142-144, March 2003.
7. W. Fang, Kang G. Shin, Dilip D. Kandlur, and D. Saha: The BLUE active queue management algorithms, IEEE/ACM Trans. On Networking, Vol. 10, No. 4, pp. 513-528, Aug. 2002.
8. S. Kunnivur and R. Srikant: Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management, Proc. Of ACM SIGCOMM 2001, San Diego, USA, August 2001, pp. 123-134.
9. J. Aweya, M. Ouellette, and D. Y. Montuno: A Control Theoretic Approach to Active Queue Management, Computer Networks, vol. 36, no. 2-3, pp. 203-235, 2001.
10. X. Deng, S. Yi, G. Kesidis, and C.R. Das: A Control Theoretic Approach for Designing Adaptive Active Queue Management Schemes, Proceedings of IEEE GlOBECOM'03, San Francisco, CA, Dec. 2003.
11. S. Ryu, C. Rump, and C. Qiao: Advances in Internet Congestion Control, IEEE Communications Surveys & Tutorials, Third Quarter 2003, vol. 5, no. 1, 2003.
12. C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong: A Control Theoretic Analysis of RED, Proceedings of IEEE /INFOCOM 2001, Anchorage, AL, April 2001.
13. G. F. Franklin, J.D. Powell, and A. Emami-Naeini, Feedback control of dynamic systems, Addsion-Wesley, 3rd ed., 1995
14. T. J. Ott, T. V. Lakshman, and L. Wong: SRED: Stabilized RED, in Proceedings of IEEE INFOCOM, vol. 3, New York, Mar. 1999, pp. 1346-1355.

15. USC/ISI, Los Angeles, CA. The NS simulator and the documentation. [Online] Available: http://www.isi.edu/nsnam/ns/
16. C. V. Hollot, Vishal Maisra, Don Towsley and Wer-Bo Gong: On designing improved controllers for AQM routers supporting TCP flows, Proc. IEEE INFOCOM, 2001, available at http://www.ieee-infocom.org/2001/paper/792.pdf
17. G. lannaccone, 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.
18. 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.