Information Sciences 180 (2010) 2249–2263

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A novel self-tuning feedback controller for active queue management supporting TCP flows

Naixue Xiong [a], Athanasios V. Vasilakos [b,*], Laurence T. Yang [c], Cheng-Xiang Wang [d], Rajgopal Kannan [e], Chin-Chen Chang [f], Yi Pan [a]

[a] Department of Computer Science, Georgia State University, USA
[b] Department of Computer and Telecommunications Engineering, University of Western Macedonia, Greece
[c] Department of Computer Science, St. Francis Xavier University, Canada
[d] Joint Research Institute for Signal and Image Processing, School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, UK
[e] Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA
[f] Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan

ARTICLE INFO

Article history:
Received 29 September 2008
Received in revised form 1 November 2009
Accepted 2 December 2009

Keywords:
Active queue management
Autonomous
Congestion control
Random early detection (RED)
Self-tuning feedback control
Stability

ABSTRACT

Wireless access points act as bridges between wireless and wired networks. Since the actually available bandwidth in wireless networks is much smaller than that in wired networks, there is a bandwidth disparity in channel capacity which makes the access point a significant network congestion point. The recently proposed active queue management (AQM) is an effective method used in wired network and wired–wireless network routers for congestion control, and to achieve a tradeoff between channel utilization and delay. The de facto standard, the random early detection (RED) AQM scheme, and most of its variants use average queue length as a congestion indicator to trigger packet dropping. In this paper, we propose a Novel autonomous Proportional and Differential RED algorithm, called NPD-RED, as an extension of RED. NPD-RED is based on a self-tuning feedback proportional and differential controller, which not only considers the instantaneous queue length at the current time point, but also takes into consideration the ratio of the current differential error signal to the buffer size. Furthermore, we give theoretical analysis of the system stability and give guidelines for the selection of feedback gains for the TCP/RED system to stabilize the instantaneous queue length at a desirable level. Extensive simulations have been conducted with ns2. The simulation results have demonstrated that the proposed NPD-RED algorithm outperforms the existing AQM schemes in terms of average queue length, average throughput, and stability.

1. Introduction

Internet congestion occurs when the aggregated demand for a resource (e.g., channel bandwidth) exceeds the available capacity of the resource. Congestion, due to the speed mismatch between a wired network and wireless LAN (WLAN) at the access point (AP), is regarded as a critical problem that affects the overall performance of WLAN. Congestion typically results in long delays in data delivery, wasted resources due to dropped packets, and the possibility of a congestion collapse [1,2,46]. Congestion control is an essential technology on the Internet, which can usually be performed by two methods: (1) by an

* Corresponding author.
E-mail addresses: nxiong@cs.gsu.edu (N. Xiong), vasilako@ath.forthnet.gr (A.V. Vasilakos), lyang@stfx.ca (L.T. Yang), cheng-xiang.wang@hw.ac.uk (C.-X. Wang), rkannan@csc.lsu.edu (R. Kannan), ccc@cs.ccu.edu.tw (C.-C. Chang), pan@cs.gsu.edu (Y. Pan).

0020-0255/$ - see front matter © 2009 Published by Elsevier Inc.
doi:10.1016/j.ins.2009.12.001

end-to-end protocol, such as TCP and (2) by an active queue management (AQM) scheme, which is implemented in routers [3,47,48] and employed to control traffic [4,5]. The basic philosophy of AQM is to trigger packet dropping (or marking, when explicit congestion notification [6,7] is enabled) before the buffer overflows, and the drop probability is proportional to the degree of congestion. AQM can achieve smaller queuing delays and higher throughput by purposely dropping packets. AQM [8,9] can be classified into two types: (1) rate-based, which controls the flow rate at the congested channel (e.g., [1]), and (2) queue-based, which controls the queue at the congested channel (e.g., [10,11]). There are several AQM schemes that have been reported in the recent literature for congestion control.

Random early detection (RED) [11–14] recommended for deployment by the Internet Engineering Task Force (IETF), is the most prominent and well-studied AQM scheme [15,16]. It is based on queue management, and has been widely implemented in routers for congestion control on the Internet. The main objective of RED is to keep the average queue length (average buffer occupancy) low. To accomplish this target, RED randomly drops incoming packets with a probability proportional to the average queue length, which makes the RED scheme suitable for irregular bursts of traffic. One important factor in measuring the performance of a traffic controller is stability: the stability of packet drop rate and the stability of queue length. A drawback of the RED method is that it is difficult to set the parameters of the RED traffic controller to stabilize the system under the diversity of Internet traffic [17,18]. The problem becomes especially severe when the average queue length reaches a certain threshold, which may result in a sharp decrease of throughput and an increase of the drop rate [3].

There are several variants of RED that have been proposed to address the above problem, such as Adaptive-RED [10,19,20], Proportional Derivative RED controller (PD-RED) [3], Proportional Integral controller (PI-RED) [21,22], and so on. With RED [11–14], the resulting average queue length is very sensitive to the level of congestion and initial parameter settings, which makes its behavior unpredictable [21]. Adaptive-RED attempts to stabilize router queue length at a level independent of the active connections, by using an exponentially weighted moving average as an integral controller [10,19,20]. Sun et al. proposed a new RED scheme [3] based on the proportional derivative control theory, called PD-RED, to improve the performance of the AQM. Unfortunately, neither Adaptive-RED nor PD-RED provides any systematic method to configure the RED parameters. Moreover, the control gain selection in both methods is based only on empirical observation and simulation analysis, which can only work well in certain given situation. A theoretical model and analysis for control gain selection and parameter setting are required. Holot et al. [21] proposed a Proportional Integral controller, PI-RED, as a means to improve the responsiveness of the TCP/AQM dynamic, and stabilize the router queue length around the target value. Similarly, Deng et al. [23] proposed a Proportional Integral Derivative model, to improve system stability under dynamic traffic conditions. Both methods used feedback control theory to describe and analyze the TCP/RED dynamic. However, the use of a highly simplified linear quadratic Gaussian controller [24] limited their discussion to the classical control elements. Consequently, their methods can only directly channel traffic control parameters to one of the AQM objectives, which leads to poor global performance. Xiong et al. proposed a Self-tuning Proportional and Integral RED (SPI-RED) [25] on average queue length, to regulate queue length. The average queue scheme keeps the average queue length low, but still allows occasional bursts of packets in the queue. In contrast, this paper proposes a novel proportional and differential control based on instantaneous queue length, called NPD-RED, which effectively avoids the problem of occasional bursts of packets.

Wireless access points act as bridges between wireless and wired networks. Since the actually available bandwidth in wireless networks is much smaller than that in wired networks, there is a bandwidth disparity in channel capacity which makes the access point a significant network congestion point. Only a handful of research papers have explored the AQM issues in WLAN. In [26], Xu et al. presented an AQM scheme for WLAN, but the performance analysis of the proposed AQM scheme was limited, because the authors only considered the delay from wired network to WLAN. On the other hand, throughput and average queue length are generally accepted as more important metrics in evaluating an AQM performance. In [27], Pang et al. mainly focused on the comparative analysis of different versions of TCPs, particularly TCP Veno [28] and TCP Reno under RED and Tail-Drop (TD) Queue. This study concluded that RED does not help in throughput in WLAN. However, it lacks the detailed analysis about the reason why RED can result in the performance degradation [29].

The main contributions of this paper are as follows. First, we propose a novel feedback control scheme, called NPD-RED, for the TCP/RED dynamic time-delayed model in wired network and wired–wireless network routers (refer to [13,30]). The core idea is new probability function for packet dropping. At the packet level, NPD-RED uses the changes in the instantaneous queue and the differential in queue length to update packet drop probability upon the arrival of new packets. On larger time-scales, NPD-RED dynamically adjusts the packet drop probability using the measured packet loss ratio. Second, we provide a theoretical analysis of the stability of the proposed probability function, and give theoretical guidance in determining the parameters for the proposed NPD-RED method. Our theoretical analysis is based on a TCP dynamic model, and uses optimal control methodologies. Furthermore, in the proposed NPD-RED method, these key parameters are decoupled from other tuning parameters, as well as from the parameters related to network conditions. The controlled parameters are adapted within dynamically changing ranges, which are determined by the stability condition. This makes the system analysis more realistic, and we propose this new configuration of NPD-RED to enhance the network performance. Finally, we conduct extensive simulations to compare the performance of our proposed method with other existing schemes. The simulation results demonstrated that the NPD-RED algorithm outperforms the existing schemes (RED, PI-RED, Adaptive-RED, and PD-RED).

The remainder of the paper is organized as follows. In Section 2, we present the system model and definitions. Section 3 discusses the NPD-RED algorithm, and develops guidelines based on theoretical analysis for choosing the parameters to achieve system stability. In Section 4, we carry out simulations under a variety of network scenarios, and analyze the performance of the proposed method. Section 5 explains more work related with AQM. Finally, we make conclusions and discuss future work in Section 6.

2. The system model and definitions

In this study, our objective is to develop an active queue management system to improve the stability of bottleneck queue in a TCP network.

In [4], a dynamic model of TCP behavior was developed, adopting fluid-flow and stochastic differential equation analysis. Simulation results demonstrated that this model accurately captured the dynamic of TCP. Following this work, a packet dropping scheme was proposed for active queue management for Internet routers in [4]. We follow the model introduced in [4]. Fig. 1 shows the theory system structure of the model for wired network and wired–wireless network, and the wired–wireless architectural trend in enterprise 802.11 deployments is shown in Fig. 2, which is included in Fig. 1.

As shown in Fig. 1, TCP sources send data packets passing through the routers to their corresponding destinations. The data will be buffered in these routers. The buffer will decide the data packet drop probability p based on the congestion of the current queue. And then it computes p to drive packet dropping. The sending window size of TCP Sender at next time slot will be adjusted based on acknowledgements of Receiver. Therefore, a closed-loop feedback model is formed (see Fig. 3).

The differential equations of the model are as follows:

$$\dot{W}(t) = -\frac{2N}{R^2C}W(t) - \frac{RC^2}{2N^2}p(t-R), \tag{1}$$

$$\dot{q}(t) = \frac{NW(t)}{R} - \frac{q(t)}{R}, \tag{2}$$

where the parameters are described in Table 1, and the round-trip time R is the sum of the processing time of buffer queue q/C and delay time of packet T_p in the channel, i.e.,

$$R = q/C + T_p.$$

Eq. (1) describes the relationship among window size dynamic, data loss probability, the number of TCP connections, the channel capacity and TCP round-trip time. And Eq. (2) shows that the change of queue length is relevant to the window size, queue length, the number of TCP connections and round-trip time.

3. The NPD-RED algorithm

In this section, we present a novel packet dropping algorithm called NPD-RED, and also give a theoretical analysis and algorithm for choosing the proportional and differential parameters to achieve system stability.

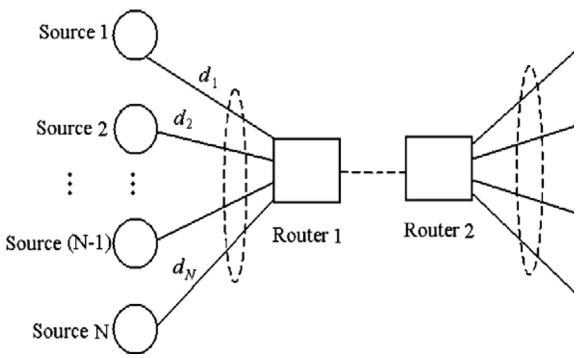

Fig. 1. The theory system network model.

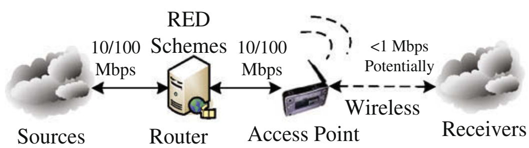

Fig. 2. Wired–wireless architectural trend in enterprise 802.11 deployments.

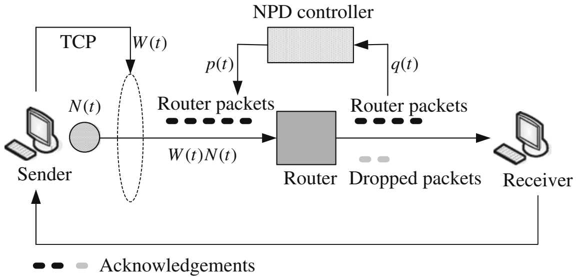

Fig. 3. The closed-loop feedback model of TCP congestion control.

Table 1
Parameters of Model (1).

Parameter	Description
W	Expected TCP window size (packets)
$\dot{W}(t)$	Time-derivative of W
q	Current queue length (packets)
C	Channel capacity (packets/second)
R	The round-trip time
N	Load factor (number of TCP connections)
p	Probability of packet dropping
$p(0)$	Initial probability of packet dropping
B	The buffer size of the congested router
q_{ref}	The expected queue length (packets)

3.1. Packet drop probability

Proportional Integral Derivative (PID) controller technology is popularly used in most industrial processes, because of its simplicity and robustness [31]. Based on our extensive experimental results and observations, we applied the classical control system techniques and came up with the design of an NPD-RED feedback controller for AQM. The new drop probability function is as follows, which is different from that in [3]:

$$p(t) = p(0) + K_p \frac{(q(t) - q_{ref})}{B} + \frac{K_d(q(t) - q(t-1))}{B}, \tag{3}$$

where $q(t)$ denotes the instantaneous queue length at time t. The component $(q(t) - q_{ref})/B$ is the instantaneous ratio of the current error signal to the buffer size, and $(q(t) - q(t-1))/B$ is the ratio of the current differential error signal to the buffer size. As you can observe from the view of theory, when the network system is completely stable, $q(t)$ converges at q_{ref}. Also, $q(t) \approx q(t-1)$, and $p(t)$ converges at $p(0)$. Therefore, in stable state, the queue length stabilizes at the expected queue length, and also drop probability is stable at a certain value. This drop probability function considers the buffer size, the current error signal, and the differential error signals. We use instantaneous queue length, instead of average queue length, as the congestion evaluation in calculating the drop probability. The reason is that the average queue scheme keeps the average queue length low, but still allows occasional bursts of packets in the queue.

The two parameters K_p and K_d need to be set in the above probability function. K_p is the proportional control gain, which is the coefficient of current error signal. K_d is the differential control gain, which is the coefficient of differential error signals. The other parameters are denoted in Table 1. As mentioned above, selecting the right control gains in the stable ranges is crucial to ensure system stability. Next, we are going to analyze system stability and provide theoretical guidelines for choosing proper control gains to optimize network performance.

3.2. Stability analysis and control gain selection

Stability in the performance of AQM scheme is essential because in a steady state, oscillations or deviations of queue length from the desired length are reduced. The main benefits of stabilizing queue length are: (1) improved utilization of resources, and (2) queuing delay reduction, by avoiding buffer overflows and oscillatory behaviors in the TCP sources. From an operational point of view, this is especially important, considering that routers have buffers of limited size. Furthermore, an unstable system often leads to strong synchronization among TCP flows. Bearing this in mind, in this paper we will consider the stabilization of queue length when setting the parameters. The parameter selection is very important to ensure the system stability. Therefore, we will analyze the proposed NPD-RED scheme based on the above TCP dynamic model, and then provide the method to ensure whole system stability.

In this section, we use the Routh–Hurwitz theorem, a common technique in control theory, to analyze the stability of our proposed model, and to determine the ranges of the control gains K_p and K_d. The stability of the system is measured by the fluctuation in queue length. The lower the fluctuation amplitude of the instantaneous queue length, the better the stability of the network system. The stability of the system effectively ensures that the instantaneous queue length converges to a certain desirable value. Because large fluctuation in queue length would lead to high packet dropping rate and poor system throughput, the stability of the congested queue length becomes an important performance metric for queue management.

To analyze the stability of the system and determine the values of the control parameters, we must obtain the characteristic equation to determine the stable range. We first linearize the above network system by performing a Laplace transform into Eqs. (1)–(3):

$$sW(s) - W(0) = -\frac{2NW(s)}{R^2C} - \frac{RC^2}{2N^2}e^{-Rs}P(s), \tag{4}$$

$$sQ(s) - q(0) = \frac{NW(s)}{R} - \frac{Q(s)}{R}, \tag{5}$$

$$sP(s) - p(0) = \frac{K_p}{B}\left(Q(s) - \frac{q_{ref}}{s}\right) + \frac{K_d}{B}(Q(s) - e^{-s}Q(s)), \tag{6}$$

where $W(s)$, $Q(s)$, and $P(s)$ denote the Laplace transform of $W(t), q(t)$, and $p(t)$, respectively. By rewriting Eqs. (4)–(6), we have:

$$W(s) = \left(W(0) - \frac{RC^2e^{-Rs}P(s)}{2N^2}\right) \Big/ \left(s + \frac{2N}{R^2C}\right), \tag{7}$$

$$Q(s) = (NW(s) + Rq(0))/(Rs + 1), \tag{8}$$

$$P(s) = \frac{p(0)}{s} - \frac{q_{ref}K_p}{Bs^2} + Q(s)\left(\frac{K_p}{Bs} + \frac{K_d}{Bs} - \frac{e^{-s}K_d}{Bs}\right). \tag{9}$$

Fig. 4 illustrates the relationships among $W(s), P(s)$ and $Q(s)$ in Eqs. (7)–(9). This basically shows how $P(s)$ transits through $W(s)$, and $Q(s)$, and eventually transits back to $P(s)$. From the diagram in Fig. 4, we can see that the system does not need any external input. It can stabilize itself purely based on internal feedback. This is, in fact, a self-tuning controller [32].

Taking $W(s), P(s)$ and $Q(s)$ as three variables in the three Eqs. (7)–(9), we solve $Q(s)$ and obtain the following characteristic function of $Q(s)$:

$$Q(s) = \left[W(0)s + \frac{Rq(0)}{N}\left(s^2 + \frac{2Ns}{R^2C}\right) + \frac{RC^2e^{-Rs}}{2N^2}\left(\frac{K_pq_{ref}}{Bs} - p(0)\right)\right] \Big/ A(s), \tag{10}$$

where

$$A(s) = \left(\frac{Rs^2}{N} + \frac{s}{N}\right)\left(s + \frac{2N}{R^2C}\right) + \frac{RC^2e^{-Rs}}{2N^2B}(K_p + K_d - K_de^{-s}). \tag{11}$$

There are several ways to test the stability of $Q(s)$ in Function (10). We employ the Routh–Hurwitz stability test [33] to determine the stability conditions of $Q(s)$. According to the control theory, the system of $Q(s)$ is stable if and only if all the zeros of $A(s)$ are in the open left half-plane (OLHP) [33]. $A(s)$ is the *characteristic polynomial* of the network system in Eq. (10) given by the original network system (1) and (2) with the controller (3). The conditions that make all the zeros of $A(s)$ in the OLHP are called stability criteria. We use the Routh–Hurwitz stability test to formulate the *stability conditions*.

Considering a general polynomial function:

$$A_N(s) = \sum_{n=0}^{N} \partial_n s^n, \quad \partial_n > 0.$$

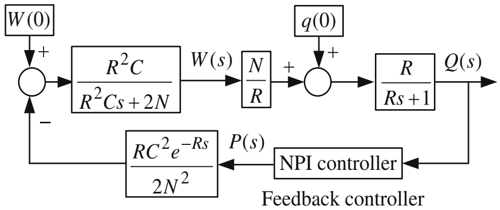

Fig. 4. The closed-loop system network model.

Table 2
The Routh array.

s^N	∂_N	∂_{N-2}	∂_{N-4}	...
s^{N-1}	∂_{N-1}	∂_{N-3}	∂_{N-5}	...
s^{N-3}	b_{N-2}	b_{N-4}	b_{N-6}	...
...	...	...	...	...
s^2	d_2	d_0	0	...
s^1	e_1	0	0	...
s^0	f_0	0	0	...

The system is stable if and only if all the solutions of s that make $A_N(s) = 0$ are inside the OLHP. The Routh–Hurwitz stability below will give the necessary and sufficient conditions for this system stability.

Given the polynomial function $A_N(s)$, we first construct the Routh array as shown in Table 2. Seen from Table 2, the first two rows of the Routh array are filled by the coefficients of $A_N(s)$, starting with the leading coefficient ∂_N. The elements in the third row are given by

$$b_{N-2} = \frac{\partial_{N-1}\partial_{N-2} - \partial_N\partial_{N-3}}{\partial_{N-1}} = \partial_{N-2} - \frac{\partial_N\partial_{N-3}}{\partial_{N-1}},$$
$$b_{N-4} = \frac{\partial_{N-1}\partial_{N-4} - \partial_N\partial_{N-5}}{\partial_{N-1}} = \partial_{N-4} - \frac{\partial_N\partial_{N-5}}{\partial_{N-1}},$$
$$\ldots$$

The elements in the fourth row are given by

$$c_{N-3} = \frac{b_{N-2}\partial_{N-3} - \partial_{N-1}b_{N-4}}{b_{N-2}} = \partial_{N-3} - \frac{\partial_{N-1}b_{N-4}}{b_{N-2}},$$
$$c_{N-5} = \frac{b_{N-2}\partial_{N-5} - \partial_{N-1}b_{N-6}}{b_{N-2}} = \partial_{N-5} - \frac{\partial_{N-1}b_{N-6}}{b_{N-2}},$$
$$\ldots$$

The other rows are computed in a similar fashion.

The Routh–Hurwitz stability test states that the system is stable (i.e., all the zeros of $A_N(s)$ are located in OLHP) if and only if all the elements in the second column of the Routh array are all strictly positive (>0). The Routh–Hurwitz test can be used to derive simple conditions for stability, expressed directly in terms of the coefficients of $A_N(s)$.

In order to compute the characteristic polynomial $A(s)$ in (11), we use the approximation $e^{-R_0 s} \approx 1 - R_0 s + R_0 s^2/2$. And based on Eq. (11), we have:

$$A(s) = s^4\left(-\frac{K_dR^3C^2}{8N^2B}\right) + s^3\left(\frac{R}{N} + \frac{K_dR^2C^2}{4N^2B} + \frac{K_dR^3C^2}{4N^2B}\right) + s^2\left(\frac{1}{N} + \frac{2}{RC} - \frac{K_dRC^2}{4N^2B} - \frac{K_dR^2C^2}{2N^2B} + \frac{K_pR^3C^2}{4N^2B}\right)$$
$$+ s\left(\frac{2}{R^2C} + \frac{K_dRC^2 - K_pR^2C^2}{2N^2B}\right) + s^0\left(\frac{K_pRC^2}{2N^2B}\right). \quad (12)$$

For simplicity, we let a_4, a_3, a_2, a_1 and a_0 denote the coefficients in the above equations of s^4, s^3, s^2, s^1 and s^0, respectively. That is:

$$a_4 = -\frac{K_dR^3C^2}{8N^2B}, \quad (13)$$
$$a_3 = \frac{R}{N} + \frac{K_dR^2C^2}{4N^2B} + \frac{K_dR^3C^2}{4N^2B}, \quad (14)$$
$$a_2 = \frac{1}{N} + \frac{2}{RC} - \frac{K_dRC^2}{4N^2B} - \frac{K_dR^2C^2}{2N^2B} + \frac{K_pR^3C^2}{4N^2B}, \quad (15)$$
$$a_1 = \frac{2}{R^2C} + \frac{K_dRC^2 - K_pR^2C^2}{2N^2B}, \quad (16)$$
$$a_0 = \frac{K_pRC^2}{2N^2B}. \quad (17)$$

Based on Eq. (12), we have:

$$A(s) = a_4s^4 + a_3s^3 + a_2s^2 + a_1s^1 + a_0s^0. \quad (18)$$

And based on the above Routh–Hurwitz stability test, we get the Routh table (Table 3).

Table 3
The Routh table.

s^4	a_4	a_2	0
s^3	a_3	a_1	0
s^2	$a_2 - \frac{a_4 a_1}{a_3}$	a_0	0
s^1	$a_1 - \frac{a_3 a_0}{\left(a_2 - \frac{a_4 a_1}{a_3}\right)}$	0	0
s_0	a_0	0	0

The network system of $Q(s)$ in (10) is stable if and only if the values in the second column of Table 3 are all greater than zero, i.e.:

$$a_4 > 0, \quad a_3 > 0, \quad a_2 - \frac{a_4 a_1}{a_3} > 0, \quad a_1 - \frac{a_3 a_0}{\left(a_2 - \frac{a_4 a_1}{a_3}\right)} > 0, \quad \text{and} \quad a_0 > 0. \tag{19}$$

We will analyze, as follows, the stability conditions given in (19) one by one.

For a_4 in (13), since $R > 0, C > 0, N > 0$, and $B > 0$, to make $a_4 > 0$, we have:

$$K_d < 0. \tag{20}$$

For a_3 in (14), to make $a_3 > 0$, we have:

$$\frac{R}{N} + \frac{K_d R^2 C^2}{4N^2 B} + \frac{K_d R^3 C^2}{4N^2 B} > 0. \tag{21}$$

Solving the above inequality, we have:

$$K_d > -\frac{4NB}{RC^2(R+1)}. \tag{22}$$

Based on Eq. (20), then we have:

$$-\frac{4NB}{RC^2(R+1)} < K_d < 0. \tag{23}$$

For $\left(a_2 - \frac{a_4 a_1}{a_3}\right) > 0$, since $a_3 > 0$, we need to get:

$$f(K_p) = a_2 a_3 - a_4 a_1 > 0. \tag{24}$$

We express Inequality (24) as a function of K_p, because we now focus on finding the range of K_p. Substitute a_1, a_2, a_3, and a_4 into (24), and obtain:

$$\left(\frac{1}{N} + \frac{2}{RC} - \frac{K_d RC^2}{4N^2 B} - \frac{K_d R^2 C^2}{2N^2 B} + \frac{K_p R^3 C^2}{4N^2 B}\right) \cdot \left(\frac{K_d R^2 C^2}{4N^2 B} + \frac{R}{N} + \frac{K_d R^3 C^2}{4N^2 B}\right) + \frac{K_d R^3 C^2}{8N^2 B} \cdot \left(\frac{2}{R^2 C} + \frac{K_d RC^2 - K_p R^2 C^2}{2N^2 B}\right) > 0,$$

i.e.,

$$\left[\frac{R^3 C^2}{4N^2 B}\left(\frac{R}{N} + \frac{K_d R^3 C^2}{4N^2 B}\right)\right] \cdot K_p > \left[\left(\frac{K_d RC^2}{4N^2 B} + \frac{K_d R^2 C^2}{2N^2 B} - \frac{1}{N} - \frac{2}{RC}\right)\left(\frac{K_d R^2 C^2}{4N^2 B} + \frac{R}{N} + \frac{K_d R^3 C^2}{4N^2 B}\right) - \frac{K_d R^3 C^2}{8N^2 B}\left(\frac{2}{R^2 C} + \frac{K_d RC^2}{2N^2 B}\right)\right]. \tag{25}$$

Inequality (25) is a linear function of K_p. While, the coefficient of K_p is

$$\frac{C^2 R^3}{4N^2 B}\left(\frac{R}{N} + \frac{K_d C^2 R^3}{4N^2 B}\right).$$

Since $C > 0, R > 0, N > 0, B > 0$, and based on Inequality (22), we have:

$$\frac{K_d C^2 R^3}{4N^2 B} > -\frac{R^2}{N(R+1)},$$

then we know:

$$\frac{R}{N} + \frac{K_d C^2 R^3}{4N^2 B} > \frac{R}{N} - \frac{R^2}{N(R+1)} > 0. \tag{26}$$

So the coefficient of K_p is positive. In order to make Inequality (24) true, the range of K_p must be:

$$K_p > \left[\left(\frac{K_dRC^2}{4N^2B}+\frac{K_dR^2C^2}{2N^2B}-\frac{1}{N}-\frac{2}{RC}\right)\left(\frac{K_dR^2C^2}{4N^2B}+\frac{R}{N}+\frac{K_dR^3C^2}{4N^2B}\right)-\frac{K_dR^3C^2}{8N^2B}\left(\frac{2}{R^2C}+\frac{K_dRC^2}{2N^2B}\right)\right]\Big/\left[\frac{R^3C^2}{4N^2B}\left(\frac{R}{N}+\frac{K_dR^3C^2}{4N^2B}\right)\right]. \tag{27}$$

For $a_1-\frac{a_3a_0}{\left(a_2-\frac{a_4a_1}{a_3}\right)}>0$, since $a_2-\frac{a_4a_1}{a_3}>0$, and $a_3>0$, we must let:

$$f(K_p)=a_1a_2a_3-a_1^2a_4-a_0a_3^2>0. \tag{28}$$

We express Inequality (28) as a function of K_p, because we now focus on finding the range of K_p. Since a_3 and a_4 are irrelevant to K_p, we substitute a_0, a_1, and a_2 into (28), and obtain:

$$\left(\frac{1}{N}+\frac{2}{RC}-\frac{K_dRC^2}{4N^2B}-\frac{K_dR^2C^2}{2N^2B}+\frac{K_pR^3C^2}{4N^2B}\right)\cdot\left(\frac{2}{R^2C}+\frac{K_dRC^2-K_pR^2C^2}{2N^2B}\right)a_3-\left(\frac{2}{R^2C}+\frac{K_dRC^2-K_pR^2C^2}{2N^2B}\right)^2a_4-\frac{K_pRC^2}{2N^2B}\cdot a_3^2>0. \tag{29}$$

Inequality (29) is a quadratic function of K_p. Furthermore, the coefficient of K_p^2 is

$$-\left(\frac{R^5C^4a_3}{8N^4B^2}+\frac{R^4C^4a_4}{4N^4B^2}\right).$$

Since $R>0, C>0, a_3>0, a_4>0, N>0$ and $B>0$, we know:

$$-\left(\frac{R^5C^4a_3}{8N^4B^2}+\frac{R^4C^4a_4}{4N^4B^2}\right)<0. \tag{30}$$

The coefficient of K_p^2 is negative. Function $f(K_p)$ in (28) is in a parabolic curve. Therefore, there are two points of K_p that make $f(K_p)=0$ and any value of $f(K_p)$ between the two points of K_p is greater than zero. Let K_{p1} and K_{p2} be the two points of K_p that make function $f(K_p)=0$, and $K_{p1}<K_{p2}$. In order to make Inequality (28) true, the range of K_p must be:

$$K_{p1}<K_p<K_{p2}. \tag{31}$$

Finally, considering a_0, we have

$$\frac{K_pRC^2}{2N^2B}>0. \tag{32}$$

Since $R>0, C>0, N>0$ and $B>0$, we know:

$$K_p>0. \tag{33}$$

We have now determined the stability condition of K_d in (23), and of K_p in (27), (31) and (33). For K_p, its ranges are given in (27), (31) and (33). We do not know which range is tighter from this analysis. The tighter range can only be obtained by computing all the inequalities out in real system situations (or in simulations). There are several methods to conduct online estimation of network parameters such as R, N, and C available in the literature. For example, the algorithm in [34] can accurately measure the round-trip time R by accepting only good samples and using the retransmission back-off strategy. Channel capacity C and TCP workload N can also be estimated according to the method proposed in [35,36].

Compared to the results reported by Hollot et al. [21] and by Low et al. [37], the stability conditions in our analysis present a clear relationship between stability and the network parameters. The analytical results provide good guidelines for choosing the important parameters of NPD-RED, leading to the desired stability and satisfactory overall performance.

3.3. The NPD-RED algorithm description

Based on the above dynamic model with the NPD-RED controller and its stability analysis, we can select the proper control gains that can ensure system stability and improve network performance. The algorithm used to compute $p(n)$ for time n (the n^{th} sampling interval) can be summarized as in Fig. 5.

Notice that after we compute the ranges of K_d and K_p, we simply choose a value for each of them randomly within their ranges. At this moment, we do not have theoretical guidance for choosing a better value. According to our simulation results, any values within the ranges make the system stable.

4. Performance evaluation

In this section, we evaluate the performance of the proposed NPD-RED packet dropping algorithm through a number of simulations performed using *NS2* [38].

a) Sample the instantaneous queue length $q(n)$;
b) Compute the range of parameters K_d by Formula (17);
c) Choose the proper value of K_d within its range;
d) Compute the range of parameter K_p by Formula (18) and the value K_d;
e) Choose the proper value of K_p within its range;
f) Compute the current drop probability $p(n)$ by Formula (3);
g) Temporarily save $p(n)$ and $q(n)$ for computation in time slot $(n+1)$.

Fig. 5. The NPD-RED algorithm.

The network topology used in the simulation is shown in Fig. 1. It is a simple dumbbell topology based on a single common bottleneck channel of 3 Mb/s capacity with identical, long-lived and saturated TCP/Reno flows. In other words, the TCP connections are modeled as greedy FTP connections, which 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 this AQM scheme, we maintain the same test conditions as in [4]: the same topology (as described above), the same saturated traffic, and the same TCP parameters.

The parameters used are explained as follows: the delay from Source i ($i \in [1, N]$) to Router 1 and from Router 2 to Receiver j ($j \in [1, N]$) are both 10 ms, and the delay between the two routers is 30 ms, then we set the round-trip time R to be 100 ms. The total buffer size is set at 75 packets, the packet size is 500 bytes, and the queue target is 50 packets. The sampling interval is set to be 0.1 ms, the number of TCP connections is considered to be 50, i.e., $N = 50$, and the total simulation time is 200 s.

4.1. How the selection of parameters influences system stability

In simulation, we observe how the selection of parameters influences system stability. Here we conduct two kinds of simulations, i.e., when control parameters are in the stable ranges (Case 1) and beyond the stable ranges (Case 2). The two cases explore how instantaneous queue length, throughput and drop probability change. Based on the above simulation results, we compute the mean queue length and the standard deviation.

According to the results of stability theory (23), (27), (31) and (33), we get: when $-3.59 \times 10^{-7} < K_d < 0, K_p > -[(0.02 - 51637.5K_d)(0.002 + 5568.75K_d) + 253.125K_d \times (4.4 \times 10^{-6} + 1.0 \times 10^5 K_d)]/(1.01 + 2.56 \times 10^5 K_d)$, and $0 < K_p < [-0.41 - 1.12 \times 10^6 K_d - 2.31 \times 10^{11} K_d^2 + 1.24 \times 10^{11} \times (1.06 \times 10^{-23} + 6.12 \times 10^{-17} K_d + 9.40 \times 10^{-11} K_d^2 + 1.59 \times 10^{-5} K_d^3 - K_d^4)^{0.5}]/(1.03 \times 10^4 + 2.59 \times 10^9 K_d)$, the system is stable; otherwise, the system is unstable. Therefore, we set $K_d = -3.59 \times 10^{-11}$, then the stable range of K_p is $0 < K_p < 3.99 \times 10^{-10}$. Now we compare the performance of the following two cases, Case 1 (stable case): $K_d = -3.59 \times 10^{-11}$ and $K_p = 2.00 \times 10^{-10}$; Case 2 (unstable case): $K_d = -3.59 \times 10^{-7}$ and $K_p = 10$.

Fig. 6 shows the change of the instantaneous queue length in the stable case. It is obvious that the queue length stabilizes quickly and there is little fluctuation at the expected value. Fig. 7 shows the dynamic change of the instantaneous queue length in the unstable case. In Fig. 7, there is a large fluctuation in the queue length, and the system is out of control. The

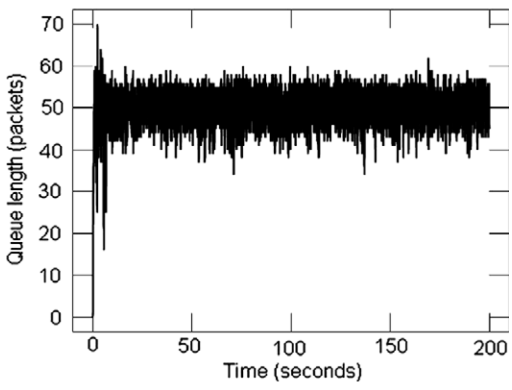

Fig. 6. Stable queue length.

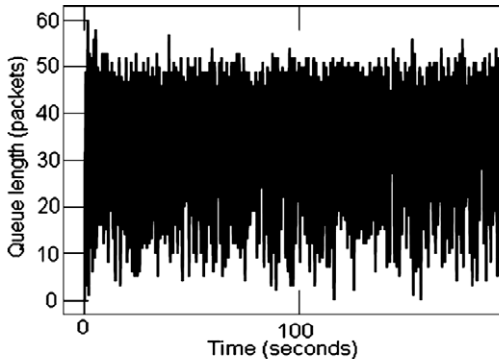

Fig. 7. Unstable queue length.

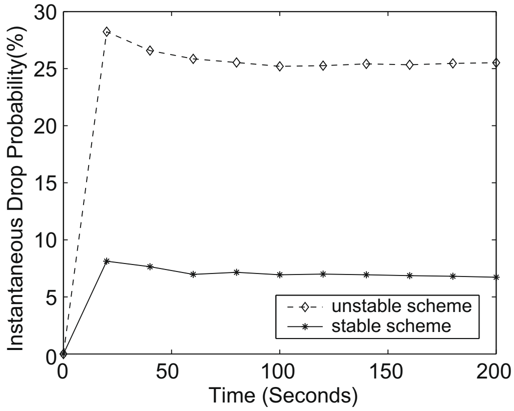

Fig. 8. Drop packet ratio comparison.

large fluctuation of queue length causes the loss of many packet, and further leads to unnecessary data retransfer, which will decrease system throughput and network quality of service (QoS).

Fig. 8 compares the packet loss ratio dynamic of the above two cases. The solid line shows the drop ratio change over time in stable case, while the dash line shows the instantaneous drop ratio dynamic over time in unstable case. Observing from these figures, the drop ratio in stable case is always less than that in unstable case.

For the above two control cases (stable case and unstable case), we compute their mean queue length and queue standard deviation. The mean queue length, in stable case, is 50.47 (close to the expected queue length), while in unstable case it is 35.17. For the queue standard deviation, it is 1.89 in stable case, while in unstable case it is 48.52. Therefore, the queue length in stable case is closer to the queue target than that in unstable case.

4.2. How the dynamic of network conditions influences the selection of control gains

Due to the continuous change of network conditions, some network parameters also change correspondingly. Choosing fixed control gains will influence system stability, and thereby influence the whole system QoS. Here we study how the network parameters (mainly considering the number of TCP connections N, and round-trip time R) influence the stable ranges of control gains. And then we give a general rule about how the network parameters influence system stability. It is better for us to adjust the control gains based on the dynamic of network conditions, so as to stabilize the system quickly and ensure the system QoS.

We consider the case in which the number of TCP connections changes, and keep the other simulation parameters unchanged (same as the above parameters, i.e., the capacity of the congestion channel is 45 Mb/s, the total buffer size is 100 packets, and the round-trip time is 100 ms). The simulation results are shown in Fig. 9. Since the upper bound of the differential parameter K_d and the lower bound of the proportional parameter K_p are zero, Fig. 9 only gives the change of the lower bound of the differential parameter K_d and the upper bound of the proportional parameter K_p. From the results, we can conclude that when the number of TCP connections increases, the lower bound of differential parameter K_d decreases linearly. However, when the number of TCP connections increases, the upper bound of the proportional parameter K_p in-

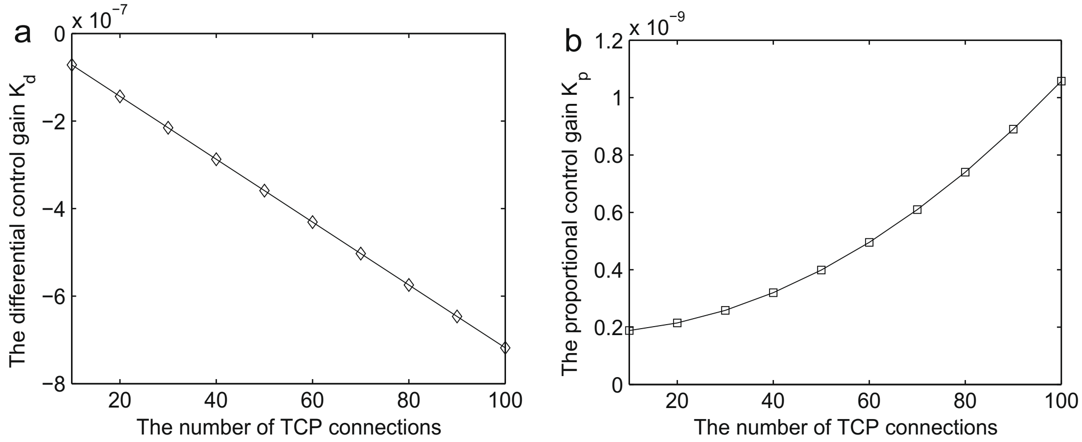

Fig. 9. Influence of the number of TCP connections on control gains.

creases in a nonlinear manner, where we set K_d to be 3.59×10^{-11}. We also simulated the influence of the number of TCP connections N to K_p with other different K_d values, and the simulation results have demonstrated the same feature as in Fig. 9(b). Therefore, in order to adjust control gains quickly to stabilize the queue length, we can first evaluate the number of TCP connections N based on the method in [35], and then select the proper control gains based on their change principle to ensure the stability of the whole system.

Furthermore, we also study the influence of round-trip time on control gains, under the conditions that the other simulation parameters remain the same (i.e., the capability of congestion channel is 3 Mb/s, the buffer size is 75 packets, and the number of TCP connections is 50). The simulation time is shown in Fig. 10. As in Fig. 9, since the upper bound of the differential parameter K_d and the lower bound of the proportional parameter K_p are zero, Fig. 10 only gives the change of the lower bound of the differential parameter K_d and the upper bound of the proportional parameter K_p. From Fig. 10, we can see that when round-trip time increases, the lower bound of the differential parameter K_d increases in a nonlinear manner, and the upper bound of the proportional parameter K_p decreases in a nonlinear manner, where we set K_d to be 3.59×10^{-11}. We also simulated the influence of the round-trip time R to K_p with other different K_d values, and the simulation results have demonstrated the same feature as in Fig. 10(b). Therefore, we can evaluate the round-trip time R based on the method in [34], and then adjust the control gains based on this above conclusion in order to make sure the system is stabilized.

From the above simulation results, we know: when control gains are set in their stable ranges, the queue length can stabilize near the expected value (as shown in Fig. 6); in contrast, when the control gains are not in their stable ranges, there is a large fluctuation (as shown in Fig. 7). Based on the statistical data, we find that the large fluctuation of queue length will lead to large data loss, which makes network throughput decrease. Furthermore, for dynamic network changes (the number of TCP connections and round-trip time), we also give the general principle of control gain change, and that principle will provide information on how to adjust the control gains. In all, it is necessary to study system stability, and to select the proper control gains for the QoS of the system.

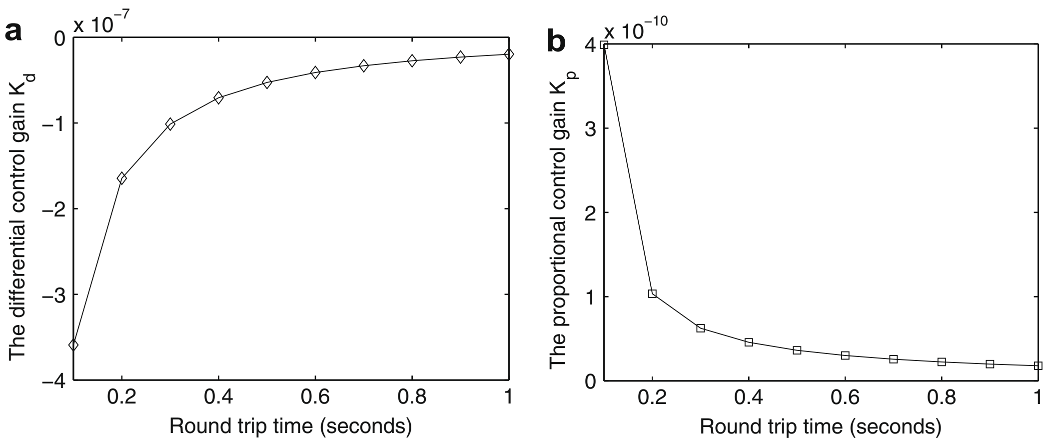

Fig. 10. Influence of TCP round-trip time on control gains.

4.3. Comparisons with the existing AQM schemes

In this simulation, we compare the performance of the proposed algorithm with the existing AQM schemes, namely, RED [13], Adaptive-RED [10], PI-RED controller [22], PD-RED [3] and NPD-RED. For all AQM schemes mentioned in this part, the simulation environment is the same as in Section 4.1.

The basic parameters of RED (see notation in [3,11,13]) are set at $interval\ time = 0.5$ s,$min_{th} = 15$ packets, $max_{th} = 75$ packets, $max_p = 0.01$ and $w_q = 0.002$, where the *interval time*, min_{th}, max_{th}, and max_p show the sampling interval time, minimum queue threshold, the maximum queue threshold and the maximum drop probability, respectively. For Adaptive-RED, the parameters are set the same as in [10]: $\alpha = 0.01, \beta = 0.9$. For PI-RED controller, PI coefficients a and b that are implemented are 1.822×10^{-5} and 1.816×10^{-5}, respectively [22]. For PD-RED, the parameters are set the same as in [3]: $\delta = 0.01, k_p = 0.001$ and $k_d = 0.05$. For NPD-RED, the parameters are same as in Section 4.1.

Figs. 11–14 and 6 show the stability of instantaneous queue dynamic for RED [13], Adaptive-RED [10], PI-RED controller [22], PD-RED [3], and NPD-RED, respectively. In Fig. 11, the experiment shows that, with RED, the queue length oscillates and fails to stabilize near the queue target of 50 packets. From Fig. 12, with Adaptive-RED, the queue stabilizes after about 15 s, so it requires long response time to stabilize the network system. In Fig. 13, the queue length of PI-RED controller stabilizes quickly, while the queue length has large fluctuations with a lot of short-lived spikes. In Fig. 14, the fluctuation amplitude of PD-RED is mostly between 30 packets and 60 packets, and the queue length fluctuates around the value 45 packets. Fig. 6 shows the stability of instantaneous queue dynamic for the presented NPD-RED, where the fluctuation amplitude of NPD-RED is mostly between 45 packets and 55 packets, and the queue length fluctuates around the target value 50 packets. These figures show that NPD-RED has better stability than RED, Adaptive-RED, PI-RED controller, and PD-RED.

The further experiment data in Table 4 compare NPD-RED with other existing AQM schemes (i.e., RED, Adaptive-RED, PI-RED controller, and PD-RED). In Table 4, it summarizes the steady-state performance, by giving, for each scheme, the average queue length and the average throughput during the whole simulation. For average queue length, the best value is the closest one to the target value 50. Thus, the average queue length 49.67 of NPD-RED is closer to the target value 50 than other

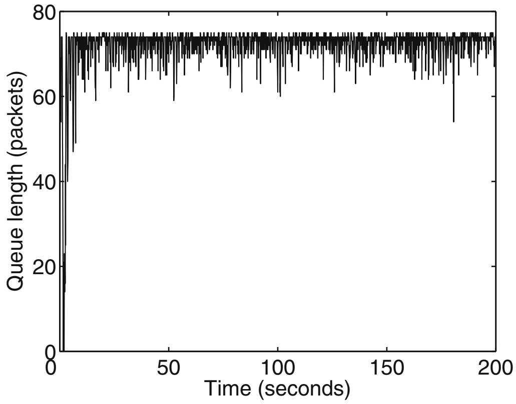

Fig. 11. Queue length: RED [13].

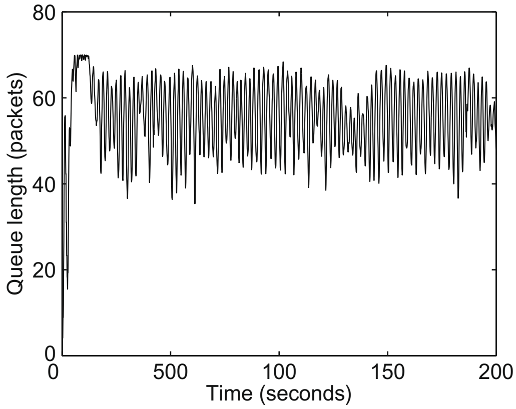

Fig. 12. Queue length: Adaptive-RED [10].

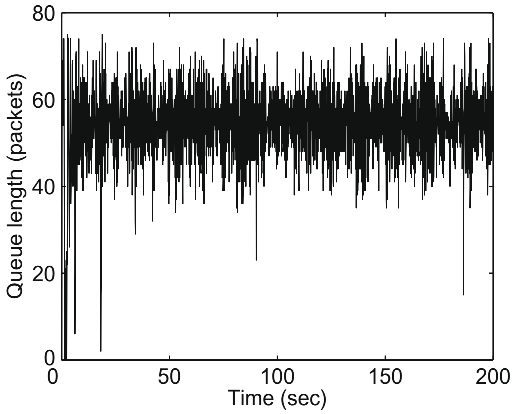

Fig. 13. Queue length: PI-RED controller [22].

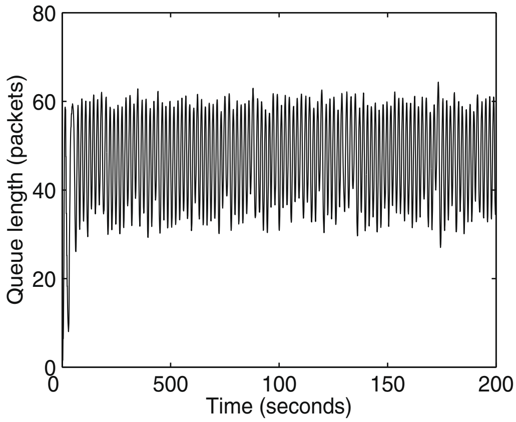

Fig. 14. Queue length: PD-RED [3].

Table 4
Simulation results comparison.

AQM scheme	RED	Adaptive-RED	PI-RED controller	PD-RED	NPD-RED
Average queue length (packets)	71.98	55.51	53.73	46.83	49.67
Average throughput (%)	89.27	84.84	85.22	93.39	93.48

four schemes. For average throughput, the larger value is better, so the average throughput 93.48 of NPD-RED is a little better than others. The comparison simulation results have demonstrated that the NPD-RED algorithm has better stability, and higher throughout than the other four schemes.

From the above simulation results, we conclude that the proposed NPD-RED scheme exhibits better network performance than RED [13], Adaptive-RED [10], PI-RED controller [22], and PD-RED [3] (in most cases).

5. Related work

In addition to the work cited in Section 1, there have been some other alternate mechanisms for AQM. For example, the Stabilized Random Early Drop (S-RED) protocol [35,39] uses adaptive methods to adjust the max drop probability p_{max}, depending on three events: buffer overflow, empty buffer and queue length increase. However, this approach introduces additional parameters that need to be configured [40].

BLUE [41] is another type of adaptive scheme. It adaptively computes packet drop probability based on only two events: buffer overflows and empty buffer. When the buffer empties (or overflows), the protocol decreases (or increases) packet drop probability by δ_2 (or δ_1). However the BLUE protocol has trouble bringing the queue length to an expected value [40]. In our paper, we use self-tuning feedback proportional and differential control theory, and not only consider the instantaneous queue length at the current time point, but also take into consideration the ratio of the current differential error signal to the buffer size to design the NPD-RED scheme, which stabilizes the instantaneous queue length at a desirable level.

Adaptive virtual queue (AVQ) [42] uses only input rate $x(t)$ to control packet dropping and to achieve expected channel utility γ, while keeping queue length small. Packet drop probability is basically proportional to the mismatch between input rate and expected channel utility γ. Through maintaining a virtual queue, AVQ deterministically drops packets upon the arrival of a new packet, realizing the same effect as probabilistic packet dropping. AVQ can achieve low average queue length and high channel utility [40], as is shown in [42]. However, as noted in [43], the rule for setting the AVQ control parameter is not scalable, because the stability condition equation in [42] becomes unsolvable as the channel capacity scales upwards, and it is due to the coupling of all the parameters. We overcome the limitation of [42] and achieve scalability by decoupling the known parameters from the control parameters. Having explicitly formulated a tractable stability range given by Formulas (17) and (18), we can make sure that the admissible control parameters are within these ranges. This has been further clarified in the above Section 2.

Random exponential marking (REM) [1] also tries to bring the queue length to an expected value. It uses the linear combination of queue mismatch and input rate mismatch to compute marking/drop probability. In REM, input rate mismatch is similarly simplified to queue variance between two continuous samplings. REM is stable for a more narrow variety of network environments than PI-RED [22] and LRED [40].

State feedback controller (SFC) [44] uses a more complete model and the TCP option of delay acknowledgment. It also uses queue mismatch and input rate mismatch as congestion indexes. SFC tries to stabilize the queue length in routers to the target value. Packet marking/drop probability in SFC is updated upon arrival of a new packet. These characterize the TCP dynamic more realistically and cause congestion window size (cwnd) decrease faster. However, SFC does not exploit Internet traffic long range dependency to design AQM [45]. Neither does it enable the controller to dynamically adapt (i.e., adapt online) to system parameter changes [25].

Loss ratio based RED (LRED) [40] measures the latest packet loss ratio, and combines it with queue length to dynamically adjust packet drop probability. However, when the network parameters are unknown, LRED can only use conservative policy to guarantee stability, and this often causes large queue deviation and lower throughput.

Misra et al. [4] discussed the difficulties in tuning RED parameters. They illustrate the benign oscillations in instantaneous queue length, and say that they are currently investigating tuning RED parameters. Hollot et al. [13] also focused on oscillations in the queue length, and use this starting point to recommend values for RED parameters. Firoiu and Borden [16] also considered problems with RED, such as oscillations in the queue length, and made recommendations for configuring RED parameters. In particular, Firoiu and Borden [16] recommended that the ideal rate for sampling the average queue length is once per round-trip time [10,25].

6. Conclusion and future work

Wireless access points act as bridges between wireless and wired networks. Since the actually available bandwidth in wireless networks is much smaller than that in wired networks, there is a bandwidth disparity between the wired and the wireless interface of an access point, which makes the access point a significant network congestion point. The recently proposed AQM is an effective method used in wired network and wired–wireless network routers for congestion control.

In this paper, we proposed a packet dropping scheme, called NPD-RED, to improve the performance of RED. We have analyzed the instantaneous queue length stability of NPD-RED, and have given guidelines for selecting control gains. This method can also be applied to other variants of RED. Based on the stability conditions and control gain selection method, extensive simulation results by NS2 demonstrate that the proposed method is effective and satisfying for guaranteeing the stability of dynamic queue, and further show the network parameters influence on stable ranges of control gains. Thus, this approach is useful in enhancing the network performance in AQM. Finally, the comparative simulation results demonstrated that the NPD-RED algorithm outperforms the existing schemes (RED, PI-RED, Adaptive-RED, and PD-RED).

Future work will cover the extension of the proposed approach from the model of a single bottleneck channel with only TCP flows, to the case of multiple bottleneck channels, as well as cases where 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-response flows will be further studied.

Acknowledgments

This research has been supported by the US National Science Foundation CAREER Award under Grant No. CCF-0545667. C.-X. Wang acknowledges the support from the Scottish Funding Council for the Joint Research Institute in Signal and Image Processing with the University of Edinburgh, as part of the Edinburgh Research Partnership in Engineering and Mathematics.

References

[1] S. Athuraliya, S.H. Low, V.H. Li, Q. Yin, REM: active queue management, IEEE Networking 15 (2001) 48–53.
[2] S.J. Lee, C.L. Hou, A neural-fuzzy system for congestion control in ATM networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 30 (1) (2000) 2–9.

[3] J. Sun, K. Ko, G. Chen, S. Chan, M. Zukerman, PD-RED: to improve the performance of RED, IEEE Communications Letters 7 (8) (2003) 406–408.
[4] V. Misra, W. Gong, D.F. Towsley, Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED, in: Proceedings of ACM SIGCOMM 2000, August 2000, pp. 151–160.
[5] M.C. Choy, D. Srinivasan, R.L. Cheu, Cooperative, hybrid agent architecture for real-time traffic signal control, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 33 (5) (2003) 597–607.
[6] J. Shapiro, C.V. Hollot, D. Towsley, Trading precision for stability in congestion control with probabilistic packet marking, in: Proceedings of 13th IEEE International Conference on Network Protocols (ICNP 2005), Boston, USA, November 6–9, 2005, pp. 118–128.
[7] S. Floyd, TCP and explicit congestion notification, ACM Computer Communication Review 24 (1994) 10–23.
[8] B. Wydrowski, M. Zukerman, QoS in best-effort networks, IEEE Communications Magazine 40 (2002) 44–49.
[9] Giuseppe Di Fatta, Frank Hoffmann, Giuseppe Lo Re, Alfonso Urso, A genetic algorithm for the design of a fuzzy controller for active queue management, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 33 (3) (2003) 313–324.
[10] S. Floyd, R. Gummadi, S. Shenker, Adaptive RED: an algorithm for increasing the robustness of RED's active queue management, Berkeley, CA, Technical Report, 2001. <http://www.icir.org/floyd/red.html>.
[11] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (1993) 397–413.
[12] C.N. Long, J. Wu, X.P. Guan, Local stability of REM algorithm with time-varying delays, IEEE Communications Letters 7 (2003) 142–144.
[13] C.V. Hollot, V. Misra, D. Towsley, W.B. Gong, A control theoretic analysis of RED, in: Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, vol. 3, April 2001, pp. 1510–1519.
[14] M. Christiansen, K. Jeffay, D. Ott, F.D. Smith, Tuning RED for web traffic, in: Proceedings of ACM SIGCOMM 2000, September 2000, pp. 139–150.
[15] S. Floyd, V. Jacobson, On traffic phase effects in packet-switched gateways, Internetworking: Research and Experience 3 (3) (1992) 115–156.
[16] V. Firoiu, M. Borden, A study of active queue management for congestion control, in: Proceedings of IEEE INFOCOM 2000, 2000, pp. 1435–1444.
[17] G. Lannaccone, M. May, C. Diot, Aggregate traffic performance with active queue management and drop from tail, ACM SIGCOMM Computer Communication Review 31 (3) (2001) 4–13.
[18] M. May, J. Bolot, C. Diot, B. Lyles, Reasons not to deploy RED, in: Proceedings of Seventh International Workshop on Quality of Service (IWQoS'99), London, UK, March 1999, pp. 260–262.
[19] M. Ouellette Aweya, D.Y. Montuno, A control theoretic approach to active queue management, Computer Networks 36 (2–3) (2001) 203–235.
[20] J. Aweya, M. Ouellette, D.Y. Montuno, A. Chapman, An optimization-oriented view of random early detection, Computer Communications 24 (2001) 1170–1187.
[21] C.V. Hollot, V. Misra, D. Towsley, W.B. Gong, Analysis and design of controllers for AQM routers supporting TCP flows, IEEE Transactions on Automatic Control 47 (2002) 945–959.
[22] C.V. Hollot, V. Maisra, D. Towsley, W. Gong, On designing improved controllers for AQM routers supporting TCP flows, in: Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, April 2001. <http://www.ieee-infocom.org/2001/paper/792.pdf>.
[23] X. Deng, S. Yi, G. Kesidis, C.R. Das, A control theoretic approach for designing adaptive active queue management schemes, in: Proceedings of IEEE GlOBECOM 2003, San Francisco, CA, December 2003. <http://www.cse.psu.edu/hpcl/papers/xdeng_globecom03.pdf>.
[24] K. Zhou, J.C. Doyle, Essentials of Robust Control, Prentice Hall, NY, USA, 1998.
[25] N. Xiong, X. Défago, X. Jia, Y. Yang, Y. He, Design and analysis of a self-tuning proportional and integral controller for active queue management routers to support TCP flows, in: Proceedings of IEEE Infocomm 2006, Barcelona, Spain, April 23–29, 2006.
[26] H. Xu, Q. Xue, A. Gam, Adaptive congestion control in infrastructure wireless LANs with bounded medium access delay, in: Proceedings of the Inremotional Mobility and Wireless Access Workshop, Fort Worth, Texas, USA, October 2002, pp. 44–49.
[27] Q. Pang, S.C. Liew, C.P. Fu, W. Wang, V.O.K. Li, Performance study of TCP Veno over WLAN and RED router, in: Proceedings of IEEE Global Telecommunications Conference, San Francisco, USA, vol. 6, December 2003, pp. 3237–3241.
[28] C.P. Fu, S.C. Liew, TCP yeno: TCP enhancement for wireless access networks, Proceedings of the IEEE Journal on Selected Areas in Communications 21 (2) (2003).
[29] S. Yi, M. Kappes, S. Garg, X. Deng, G. Kesidis, C.R. Das, Proxy-RED: an AQM scheme for wireless local area networks, in: Proceedings of 13th International Conference on Computer Communications and Networks (ICCCN 2004), October 11–13, 2004, pp. 460–465.
[30] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, in: Proceedings of ACM SIGCOMM, Vancouver, CA, August 1998, pp. 303–314.
[31] E. Harinath, G.K.I. Mann, Design and tuning of standard additive model based fuzzy PID controllers for multivariable process systems, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38 (3) (2008) 667–674.
[32] A. Rogers, E. David, N.R. Jennings, Self-organized routing for wireless microsensor networks, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 35 (5) (2005) 349–359.
[33] W. Kamen, B.S. Heck, Fundamentals of Signals and Systems Using the Web and Matlab, 2th ed., Prentice Hall, Inc., 2002. pp. 439–443.
[34] P. Karn, C. Partridge, Improving round-trip time estimates in reliable transport protocols, ACM Computer Communication Review 25 (1) (1995) 66–74.
[35] T.J. Ott, T.V. Lakshman, L. Wong, SRED: stabilized RED, in: Proceedings of IEEE INFOCOM, NY, vol. 3, March 1999, pp. 1346–1355.
[36] H. Zhang, C.V. Hollot, D. Towsley, V. Misra, A self-tuning structure for adaptation in TCP/AQM networks, ACM Sigmetrics Performance Evaluation Review 31 (1) (2003) 302–303.
[37] S.H. Low, F. Paganini, J. Wang, J.C. Doyle, Linear stability of TCP/RED and a scalable control, Computer Networks Journal 43 (5) (2003) 633–647.
[38] USC/ISI, Los Angeles, CA, The NS Simulator and the Documentation. <http://www.isi.edu/nsnam/ns/>.
[39] W. Feng, D.D. Kandlur, D. Saha, K.G. Shin, A self-configuring RED gateway, in: Proceedings of IEEE INFOCOMM 1999, New York, NY, vol. 3, March 21–25, 1999, pp. 1320–1328.
[40] C. Wang, B. Li, Y. Thomas Hou, K. Sohraby, Y. Lin, LRED: a robust active queue management scheme based on packet loss ratio, in: Proceedings of IEEE INFOCOM 2004, Hong Kong, March 7–11, 2004. <http://www.ieee-infocom.org/2004/Papers/01_1.PDF>.
[41] W. Fang, Kang G. Shin, D.D. Kandlur, D. Saha, The BLUE active queue management algorithms, IEEE/ACM Transactions on Networking 10 (4) (2002) 513–528.
[42] S. Kunniyur, R. Srikant, Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management, in: Proceedings of ACM SIGCOMM, San Diego, CA, August 2001, pp. 123–134.
[43] D. Katabi, C. Blake, A note on the stability requirements of adaptive virtual queue, LCS Document Number: MIT-LCS-TM-626, February 13, 2002. <http://www.lcs.mit.edu/publications/pubs/pdf/MITLCSTM626.pdf>.
[44] Y. Gao, J.C. Hou, A state feedback control approach to stabilizing queues for ECN-enabled TCP flows, in: Proceedings of IEEE INFOCOM 2003, San Francisco, CA, vol. 3, March 30–April 2, 2003, pp. 2301–2311.
[45] Y. Gao, G. He, J. Hou, On leveraging traffic predictability in active queue management, in: Proceedings of IEEE INFOCOM 2002, New York, USA, June 23–27, 2002. <http://lion.cs.uiuc.edu/courses/cs497hou/PAQM.pdf>.
[46] E.N. Huh, H. Choo, Performance enhancement of TCP in high-speed networks, Information Sciences 178 (2) (2008) 352–362.
[47] C.K. Chen, Y.-C. Hung, T.-L. Liao, J.-J. Yan, Design of robust active queue management controllers for a class of TCP communication networks, Information Sciences 177 (19) (2007) 4059–4071.
[48] H.S. Tsai, W.K. Lai, Improving fairness among TCP flows by stateless buffer control with early drop maximum, Information Sciences 178 (19) (2008) 3697–3715.