Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 36 (2009) 1903–1913

Expert Systems with Applications

www.elsevier.com/locate/eswa

GA-based PID active queue management control design for a class of TCP communication networks

Chang-Kuo Chen [a], Hang-Hong Kuo [a], Jun-Juh Yan [b], Teh-Lu Liao [a,*]

[a] Department of Engineering Science, National Cheng Kung University, No. 1 University Road, Tainan 701, Taiwan, ROC
[b] Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan, ROC

Abstract

Active queue management (AQM) is a key congestion control scheme for reducing packet loss and improving network utilization in TCP/IP networks. This paper proposes a proportional–integral-derivative (PID) controller as an active queue manager for Internet routers. Due to the limitations of packet-dropping probability and the effects of propagation delays in TCP networks, the TCP AQM network was modeled as a time-delayed system with a saturated input. An improved genetic algorithm is employed to derive optimal or near optimal PID controller gains such that a performance index of integrated-absolute error (IAE) is minimized, and thereby a stable queue length, low packet loss, and high link utilization for TCP networks are guaranteed. The performance of the proposed control scheme is evaluated in various network scenarios via a series of numerical simulations.

Keywords: AQM; TCP/IP; Congestion control; Genetic algorithm; PID controller

1. Introduction

Since the unpredictable interference and the number of users has grown rapidly in today's Internet environment, network traffic congestion results in long time delays for data transmission and frequently makes the queue length in the buffer of the intermediate router (switch) overflow, and can even lead to network collapse (Huang, Cheng, Chuang, & Jang, 2006; Kim et al., 2007). Therefore, a congestion control mechanism which determines a queuing guideline for the routers (switches) should be proposed to indicate the order in which packets are delivered and denote which packets should be dropped when congestion occurs. The active queue management (AQM) scheme is an efficient solution that detects inceptive congestion and gives early notice of conditions of the information for the current Internet situation by dropping (or marking) incoming packets before router queues become full.

The first well-known AQM scheme, random early detection (RED) (Floyd & Jacobson, 1993) algorithm, was developed and introduced into Internet routers for reducing the flow synchronization problem and clamed the traffic load via measurement of average queue length. Nevertheless, several studies and theoretical analyses have shown that the performance of RED is sensitive to its parameter settings and traffic load due to it being designed in an *ad hoc* architecture (Feng, Shin, Kandlur, & Saha, 2002). Therefore, a number of new modified schemes including ARED (Floyd, Gummadi, & Shenker, 2001), FRED (Lin & Morris, 1997), and SRED (Qtt, Lakshman, & Wong, 1999), have been proposed in the literature. However, those studies are unable to maintain the system performance in a wide range of operational conditions such as the number of connections, propagation delay, and link capacity.

A fluid-based model of the dynamics of the TCP and RED was developed by the stochastic theory (Misra, Gong, & Towsley, 2000). This model represents the behavior of the characteristic variables of the network and shows that it accurately captured the qualitative evolution of TCP traffic flows. Based on this TCP model, several congestion

* Corresponding author. Tel.: +886 6 2757575; fax: +886 6 2766549.
E-mail address: tlliao@mail.ncku.edu.tw (T.-L. Liao).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.12.003

control schemes have been developed to improve the performance of communication networks (Ren, Lin, & Yin, 2005; Wang & Hollot, 2003; Yan, Gao, & Özbay, 2003; Zhang, Liu, & Dou, 2003).

On the other hand, based on the linearized fluid-based TCP/AQM model, a proportional–integral (PI) controller (Hollot, Misra, Towsley, & Gong, 2001) was developed to regulate the queue level, round-trip time, and packet loss. However, PI controller is sluggish with taking too long time to achieve the desired queue length due to PI controller improves the steady-state error at the cost of an increase in rise time (Bequette, 2003). Therefore, in order to overcome the drawbacks of PI controller, several new modified methods are proposed. The virtual rate control (VRC) algorithm for AQM in TCP networks has been proposed in Park, Lim, Park, and Choi (2004). In Ren, Lin and Wei (2005), a DC-AQM algorithm based on PID control was developed for large delay networks. A Smith predictor-based PI (SPPA) controller was presented to reduce the effects of the time delays in the control loop by a predictor (Li, Ko, & Chen, 2005). In the work of Wang, Wang, Zhou, and Huai (2006), the Particle Swarm optimization (PSO)-based PID controller was presented for an improved TCP/AQM with large-delay network situations. For improving the transient performance of the fixed-gain PI controller over a wide range of uncertainties, a stable queue-based adaptive proportional–integral (Q-SAPI) controller for AQM was proposed in Chang and Muppala (2006). However, the above researches are based on the linearized model and the complexity of the controller design, except for PID or PI controllers, is hard to realize. In addition, a saturated nonlinearity of the control input usually exists in this control problem due to the property of packet-dropping probability. Therefore, the effect of a saturated actuator should be considered; otherwise it can cause serious degradation and instability in a network with typically large-scale, complex systems. Accordingly, based on PID control, an efficient selection mechanism of the parameters for the PID controller with the property of input saturation, in this complex nonlinear model should be anticipated.

The Genetic algorithm (GA) has been considered as a useful technique employing the principles of natural genetic systems (Goldberg, 1989) to search a global solution of optimization problem. The basic idea is to maintain a population of possible solution that evolves and improves over time through a process of competition and controlled variation. It has been successfully applied in different areas such as modeling and classification (Setnes & Roubos, 2000), power quality assessment (El-Zonkoly, 2005), resource allocation (Wang & Lin, 2007) and, adaptive scheduling system (Juang, Lin, & Kao, 2007), etc.

In this paper, we develop a PID controller for a time-delayed TCP system with input saturation to ensure stable queue length, low packet loss, and high link utilization. Based on the improved GA search method, an efficient mechanism for selecting the controller gains of the PID controller is proposed. The proposed GA-based PID AQM scheme possesses a reliable stability and is robust against the number of TCP sessions, the various RTT, bursting and unresponsive flows, etc.

The rest of this research is organized as follows. Section 2 presents the TCP model and the control objective. An improved GA-based PID controller design for AQM is illustrated in Section 3. Section 4 shows the results of simulations to demonstrate the performance of the proposed control scheme. Finally, brief conclusions are provided in Section 5.

2. System description and problem formulation

A window-based nonlinear fluid-flow dynamic model for TCP networks is considered in this study. A detailed justification of this model was presented in Hollot et al. (2001) and Kelly (2001). Briefly, this model expresses the coupled non-linear differential equations such that they reflect the dynamics of TCP accurately with the average TCP window size and the average queue length. The coupled non-linear differential equations are given as follows:

$$\dot{w}(t) = \frac{1}{\frac{q(t)}{C} + T_p} - \frac{w(t)}{2}\frac{w(t-R(t))}{\frac{q(t-R(t))}{C} + T_p}p(t-R(t)) \tag{1.a}$$

$$\dot{q}(t) = \begin{cases} -C + \frac{N(t)}{\frac{q(t)}{C}+T_p}w(t) & \text{if } q(t) > 0 \\ \max\left\{0, -C + \frac{N(t)}{\frac{q(t)}{C}+T_p}w(t)\right\} & \text{if } q(t) = 0 \end{cases} \tag{1.b}$$

where w is the average TCP window size (in packets); q is the instantaneous queue length (in packets); T_p is the propagation delay (in seconds); R is the transmission RTT, equal to $q/C + T_p$; C is the link capacity (in packets/sec); N is the number of TCP connections; and p is the packet-dropping probability, which is the control input to decrease the sending rate and maintain the bottleneck queue length. All of the above variables are supposed to be nonnegative. In Eq. (1.a), the additive increase and multiplicative decrease (AIMD) congestion control algorithm is used to evaluate the average window size during the TCP flow, while Eq. (1.b) is the dynamics of the queue length accumulated as the transmission rate surpasses the link capacity.

Since the packet-dropping probability is between 0 and 1, the following nonlinear time-delayed system with a saturated input can be derived from Eq. (1):

$$\dot{w}(t) = \frac{1}{\frac{q(t)}{C} + T_p} - \frac{w(t)}{2}\frac{w(t-R(t))}{\frac{q(t-R(t))}{C} + T_p}\mathrm{sat}(u(t)) \tag{2.a}$$

$$\dot{q}(t) = \begin{cases} -C + \frac{N(t)}{\frac{q(t)}{C}+T_p}w(t) & \text{if } q(t) > 0 \\ \max\left\{0, -C + \frac{N(t)}{\frac{q(t)}{C}+T_p}w(t)\right\} & \text{if } q(t) = 0 \end{cases} \tag{2.b}$$

The saturated input $u(t) = p(t - R(t))$ is expressed by the following nonlinearity:

$$sat(u(t)) = \begin{cases} u_{\max}, & u(t) \geqslant u_{\max} \\ u(t), & u_{\min} \leqslant u(t) < u_{\max} \\ u_{\min}, & u(t) < u_{\min} \end{cases} \tag{3}$$

where the lower boundary and upper boundary are given as $u_{\min} = 0$ and $u_{\max} = 1$.

In this study, an improved GA-based PID controller is proposed to achieve the desired queue length efficiently and provide robust performance with delay effects and a saturated input. A PID controller generates the term $u(t)$ as a control input in Eq. (2.a) to guarantee the stability of system (2). Furthermore, the output error signal is defined as $e(t) = q(t) - q_{\mathrm{d}}$, where q_{d} denotes the desired queue length. In application, a PID controller with an input $e(t)$ and an output $u(t)$ is expressed as follows:

$$u(t) = K_{\mathrm{P}}\left[e(t) + \frac{1}{T_{\mathrm{I}}}\int_0^t e(\tau)\,\mathrm{d}\tau + T_{\mathrm{D}}\frac{\mathrm{d}}{\mathrm{d}t}e(t)\right] \tag{4}$$

where K_{P} is the proportional gain, T_{I}, and T_{D} are the integral time and the derivative time constants, respectively. Similarly, Eq. (4) can be also rewritten as

$$u(t) = K_{\mathrm{P}}e(t) + K_{\mathrm{I}}\int_0^t e(\tau)\,\mathrm{d}\tau + K_{\mathrm{D}}\frac{\mathrm{d}}{\mathrm{d}t}e(t) \tag{5}$$

where $K_{\mathrm{I}} = K_{\mathrm{P}}/T_{\mathrm{I}}$ is the integral gain and $K_{\mathrm{D}} = K_{\mathrm{P}}T_{\mathrm{D}}$ is the derivative gain.

In order to measure the performance of the closed-loop control system, an integral absolute error (IAE) is employed as the objective function (OF)

$$\mathrm{OF} := \mathrm{IAE} = \int_0^\infty |e(\tau)|\,\mathrm{d}\tau \tag{6}$$

In fact, the objective function will depend on the controller gains K_{P},K_{I}, and K_{D}. The smaller IAE is get, the better PID controller is designed. Practically, the design of the PID controller gain is very heuristic and depends on expert's experiences. In the next section, an improved GA-based searching method determining the optimal gains of PID controller to minimize the objective function will be presented.

3. An improved GA-based PID controller design

Genetic algorithm is an artificial optimization scheme developed in analogy to natural evolution performing an exploration of the search space. It has been considered as an efficient technique for searching the global or near global solution of complex optimization problems. Therefore, an improved GA-based search method is proposed here to find the optimal values of the PID controller gains K_{P}, K_{I}, and K_{D} for the AQM to support the TCP such that the value of IAE in (6) is minimal.

3.1. Optimization procedures

The procedures of standard GA for solving the above optimization problem are described as follows (Goldberg, 1989):

Step 1: Generate random population strings (chromosomes).
Step 2: Calculate the fitness for each string in the population.
Step 3: Create offspring using genetic operators by initially employing roulette wheel selection, then crossover, and finally mutation.
Step 4: Stop if the search goal is achieved. Otherwise repeat with Step 2.

In the application, the first step involves encoding the values of K_{P}, K_{I}, and K_{D} into a binary string of fixed-length. Without loss of generality, assume that there are M_1, M_2, and M_3 bits for each values of K_{P}, K_{I}, and K_{D}, respectively. Therefore, $S = [K_{\mathrm{P}}K_{\mathrm{I}}K_{\mathrm{D}}]$ is transformed into a string (chromosome) that has $(M_1 + M_2 + M_3)$ bits as follows:

$$S = \left[\underbrace{b_{11}, b_{i2}, \ldots, b_{1M_1}}_{s_1=K_{\mathrm{P}}}\ \underbrace{b_2, b_{22}, \ldots, b_{2M_2}}_{s_2=K_{\mathrm{I}}}\ \underbrace{b_{31}, b_{32}, \ldots, b_{3M_3}}_{s_3=K_{\mathrm{D}}}\right] \tag{7}$$

The corresponding values of K_{P}, K_{I}, and K_{D} are determined as follows:

$$s_j = L_j + D_j\frac{(U_j - L_j)}{2^{M_j} - 1} \tag{8}$$

where D_j is the decimal number which converts from population string s_j; U_j and L_j denote the lower boundary and upper boundary of search ranges, respectively, $j = 1, 2, 3$.

As a second step, a fitness function $f(S)$ related to the objective function OF as defined in Eq. (6) is given by

$$f(S) = \frac{1}{1 + h(S)} = \frac{1}{1 + \int_0^\infty |e(\tau)||_{S=[K_{\mathrm{P}},K_{\mathrm{I}},K_{\mathrm{D}}]}\,\mathrm{d}\tau} \tag{9}$$

The smaller $h(S)$ is; the larger $f(S)$ is. However, the values of fitness function falling in the small range will possess a better classification performance. Therefore, the following normalized fitness function is proposed here:

$$f'(S) = \frac{f(S) - f_{\min}}{f_{\max} - f_{\min}} \tag{10}$$

where $f_{\max}$ and $f_{\min}$ present the maximum and minimum fitness value in a generation, respectively.

Traditionally, the parameters of GA are chosen by manually tuning for each search problem, as large values of crossover probability p_{c} (0.6–0.9) and the small mutation probability p_{m} (0.01–0.1) are frequently used in GA practice (Johnson & Rahmat-Samii, 1997). In fact, identifying optimal settings for p_{c} and p_{m} is a complex task. The adaptive genetic algorithm (AGA) scheme was proposed to adjust p_{c} and p_{m} by using the value of $f_{\max} - f_{\mathrm{avg}}$ at each generation in Srinivas and Patnaik (1994), and $f_{\mathrm{avg}} = \frac{1}{N_{\mathrm{P}}}\sum_{i=1}^{N_{\mathrm{P}}} f_i$ denotes the average fitness value, where N_{P} is the number of population in a generation. The adaptive strategy for varying p_{c} and p_{m} is expressed as follows:

$$p_c = \begin{cases} k_1(f_{\max} - f_c)/(f_{\max} - f_{avg}) & \text{if } f_c \geqslant f_{avg} \\ k_3 & \text{if } f_c < f_{avg} \end{cases} \tag{11}$$

and

$$p_m = \begin{cases} k_2(f_{\max} - f_m)/(f_{\max} - f_{avg}) & \text{if } f_m \geqslant f_{avg} \\ k_4 & \text{if } f_m < f_{avg} \end{cases} \tag{12}$$

where f_c denotes the maximal fitness value of the individuals selected for crossover; f_m corresponds to the fitness values of the solutions to be mutated; and k_1, k_2, k_3, and k_4 are constant values and less than 1.0 to constrain p_c and p_m into the range $[0, 1]$.

However, both p_c and p_m are always zero when the optimal solution with the maximum fitness value occurs. Therefore, a new efficient scheme of p_c and p_m to improve these problems is proposed as follows:

$$p_c = \begin{cases} (k_1(f'_{\max} - f'_c)/(f'_{\max} - f'_{avg} + \delta)) & \\ \quad \times a + p_{c\_last} \times (1 - a) & \text{if } f'_c \geqslant f'_{avg} \\ k_3 \times a + p_{c\_last} \times (1 - a) & \text{if } f'_c < f'_{avg} \end{cases} \tag{13}$$

and

$$p_m = \begin{cases} (k_2(f'_{\max} - f'_m)/(f'_{\max} - f'_{avg} + \delta)) & \\ \quad \times a + p_{m\_last} \times (1 - a) & \text{if } f'_m \geqslant f'_{avg} \\ k_4 \times a + p_{m\_last} \times (1 - a) & \text{if } f'_m < f'_{avg} \end{cases} \tag{14}$$

where f' denotes the modified objective function which normalizes the fitness value by Eq. (10); p_{c_last} and p_{m_last} are crossover and mutation probabilities of the last generation, respectively; a denotes a weight of probability; δ is a small constant value such that the first of (13) and (14) are well defined. Meanwhile, we take the strategy of replacing worse strings of the new population with the best string of the current population as an elitism strategy.

The details of the proposed GA-based PID controller design are described as follows:

Step 1: Define the search range of the values of K_P, K_I, and K_D, and determine the parameters of M_1, M_2, and M_3 for K_P, K_I, and K_D.

Step 2: Define the fitness function as given in Eq. (10).

Step 3: Determine the population size N_P and generate N_P populations $S_1, S_2, \ldots, S_{N_p-1}$ and S_{N_p} all having $(M_1 + M_2 + M_3)$ bits by the quasi-random sequence (QRS) (Cao, 1997) to obtain the initial population $P^0 = \{S_1, S_2, \ldots, S_{N_p}\}$.

Step 4: Calculate the fitness function for each S_i from the kth generation P^k, and then find the best string S_i^* of P^k such that $f'(S_i^*) = \max_{S_i \in P^k}\{f'(S_i)\}$. If the best strings are not unique, then set any one of the best strings in P^k as S_i^*.

Step 5: Construct the mating pool and calculate the parameters p_c and p_m by Eqs. (13) and (14). Perform crossover and mutation operations on the strings S_i of the mating pool and obtain a new population S_i^{new}.

Step 6: Compare the fitness function of each S_i with that of S_i^{new}. If $f'(S_i^{new}) < f'(S_i)$, then replace the string of S_i^{new} by S_i, and then put it into the new generation. Otherwise, put S_i^{new} into the new generation.

Step 7: Repeat steps 4–6 iteratively until the value of $f'(S_i^*)$ is converged.

Step 8: Set S^* by S_i^* and decode the chromosome S^* (represented by a binary string) having the maximum fitness value into its corresponding values of K_P^*, K_I^*, and K_D^*.

Step 9: Stop the algorithm.

4. Simulation results

In this section, we illustrate the GA-PID controller design for TCP/AQM. For a TCP/AQM network modeled by Eq. (2), it is assumed that $N = 100$ homogeneous TCP connections and shares one bottleneck link with a capacity of 10 Mbps, i.e. $C = 1250$ (packets/second). Furthermore, the propagation delay of the bottle link capacity was $T_p = 0.08$, the desired queue size was $q_d = 150$ packets and, therefore, $u_{\min} = 0$ and $u_{\max} = 1$. In the improved GA operation, the number of population is set as $N_p = 50$, and the initial population is $P^0 = \{S_1, S_2, \ldots, S_{N_P}\}$ in which each S_i has the 30 bits string length (i.e. $M_j = 10$, $j = 1, 2, 3$) and is generated by QRS, and the search ranges of K_P, K_I, and K_D are set as $U_j = 0$ and $L_j = 0.1$ for all $j = 1, 2, 3$, respectively. Moreover, the initial conditions of crossover and mutation probabilities are selected as $p_c^0 = 0.8$ and $p_m^0 = 0.05$ as well as the constant parameters are given by $k_1 = k_3 = 0.8$, $k_2 = k_4 = 0.3$, $a = 0.6$, and $\delta = 0.01$. After a series of improved GA manipulations, the convergence curve of the IAE value vs. iteration is described in Fig. 1, and its final value is $f(S^*) = 0.6644$. Correspondingly, the GA-PID control gains were obtained as $K_P^* = 2.4869 \times 10^{-6}$, $K_I^* = 2.2233 \times 10^{-8}$, and $K_D^* = 6.0046 \times 10^{-5}$.

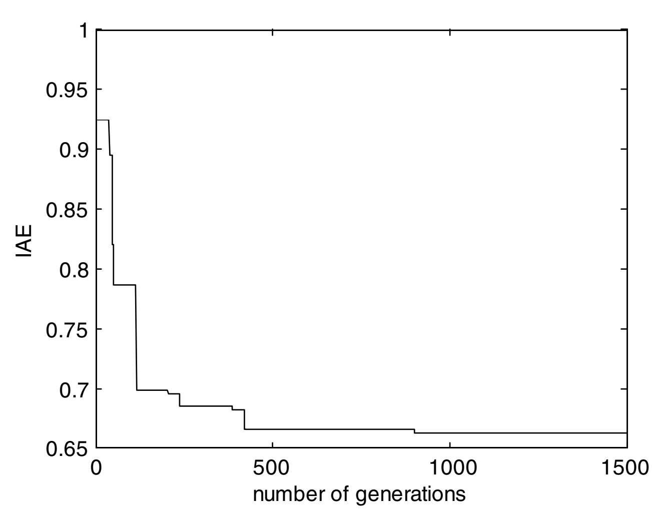

Fig. 1. Convergence of IAE value vs. iteration.

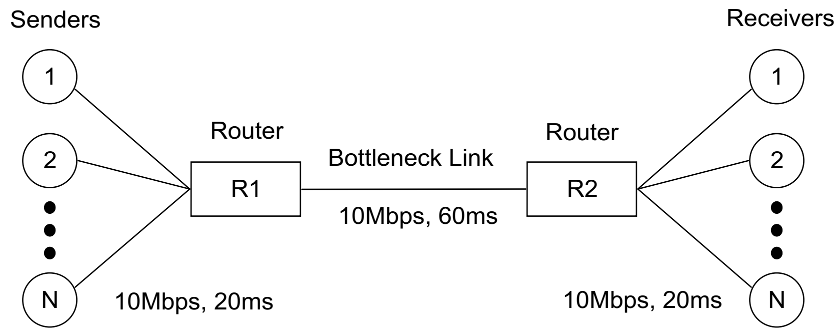

Fig. 2. Dumbbell network topology.

The performance and effectiveness, such as high throughput, predictable delay and link utilization of the proposed GA-PID controller, were verified in a series of numerical simulations using NS-2 (Network Simulator-2) with the dumbbell network topology as shown in Fig. 2. In this topology, the transport agent is based on TCP-Reno, where multiple TCP connections share a single bottleneck link. Each link capacity, as well as the propagation delay, is depicted in Fig. 2. It was supposed that the TCP sources sent their data incessantly. Unless otherwise noted, the maximum buffer size of each router was set at 300 packets, where each packet had a size of 1000 bytes. The objective queue length q_d is set to 150 packets. To demonstrate the robustness of the proposed AQM controller, we took into account the dynamic traffic changes, different RTTs, bursting and unresponsive flows, and multiple bottleneck links in the simulated network in further simulations. The simulation results for these conditions using the proposed control strategy were then compared with other AQM schemes, such as PI (Hollot et al., 2001), ARED (Floyd et al., 2001), and REM (Athuraliya, Low, Li, & Yin, 2001). The parameters of each AQM are listed in Table 1.

Table 1
The parameter setting of PI, ARED, REM, and GA-PID controllers.

AQM scheme	Parameters
PI	$a = 1.816 \times 10^{-5}$, $b = 1.822 \times 10^{-5}$, $T = 1/160$ s
ARED	$\min_{th} = 0.8 \times q_d$, $\max_{th} = 1.5 \times q_d$, $w_g = 1 - \exp(-1/C)$
REM	$\gamma = 0.001$, $\phi = 1.001$
GA-PID	$K_P = 2.4869 \times 10^{-6}$, $K_I = 2.2233 \times 10^{-8}$, $K_D = 6.0046 \times 10^{-5}$, $T = 1/160$ s

4.1. Performance of constant TCP connections

In this simulation experiment, the results obtained with different AQM schemes are shown in Fig. 3. It is clear that the presented GA-PID controller and ARED successfully achieve the desired queue length and rapidly stabilize the network at the operating point. Although the transient

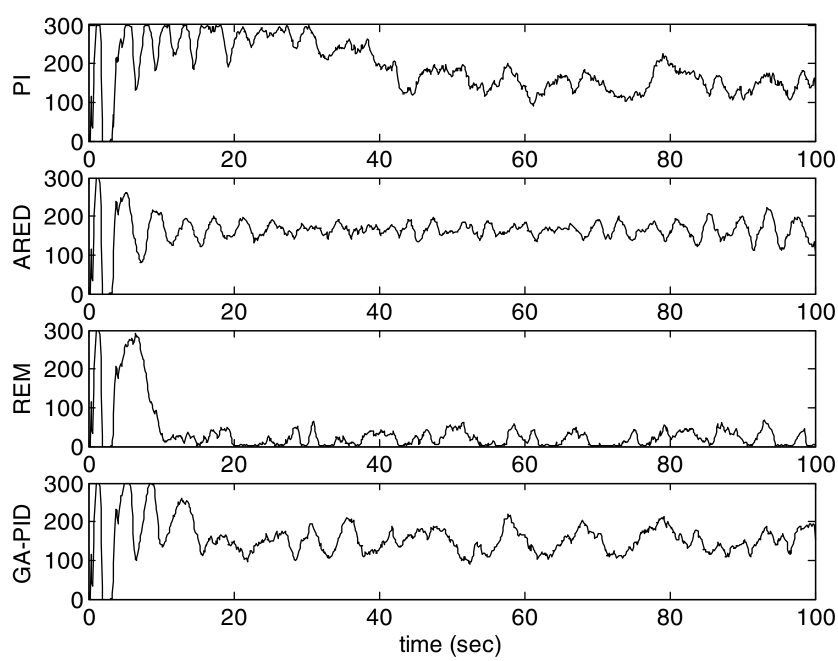

Fig. 3. Responses of queue length (in packets) on dumbbell network topology.

responses of PI can maintain the queue length around the desired value, too inactive and serious overshoots occurred in the PI schemes. However, the REM scheme cannot maintain the desired queue length because there is no parameter of the target queue length in its control mechanism.

4.2. Performance of dynamic traffic load

Dynamic traffic changes were considered in this experiment. In this situation, 100 TCP-Reno connections were enabled at time $t = 0$. At time $t = 30$, another 50 TCP connections begin transmission and remained inactive until time $t = 60$. Additionally, 100 TCP connections departed at the same time. Until time $t = 80$, 50 TCP connections reinitiated sending data. Fig. 4 shows the corresponding queue evolutions obtained under the various AQM schemes. It can be seen that both PI and ARED controllers are not robust with respect to variations in the load. Nevertheless, the proposed GA-PID is robust to variations in the number of active connections.

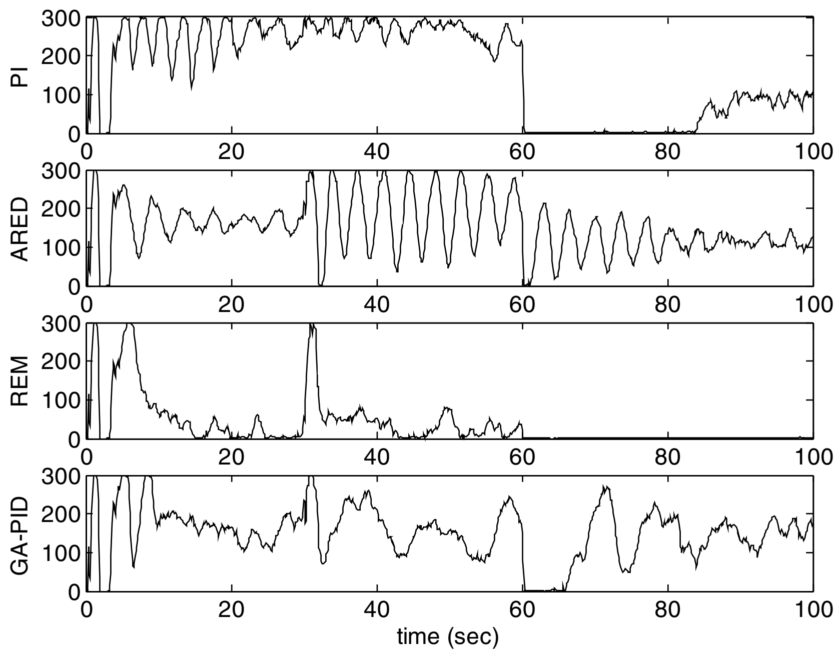

Fig. 4. Responses of queue length (in packets) on dumbbell network topology subject to dynamic traffic changes.

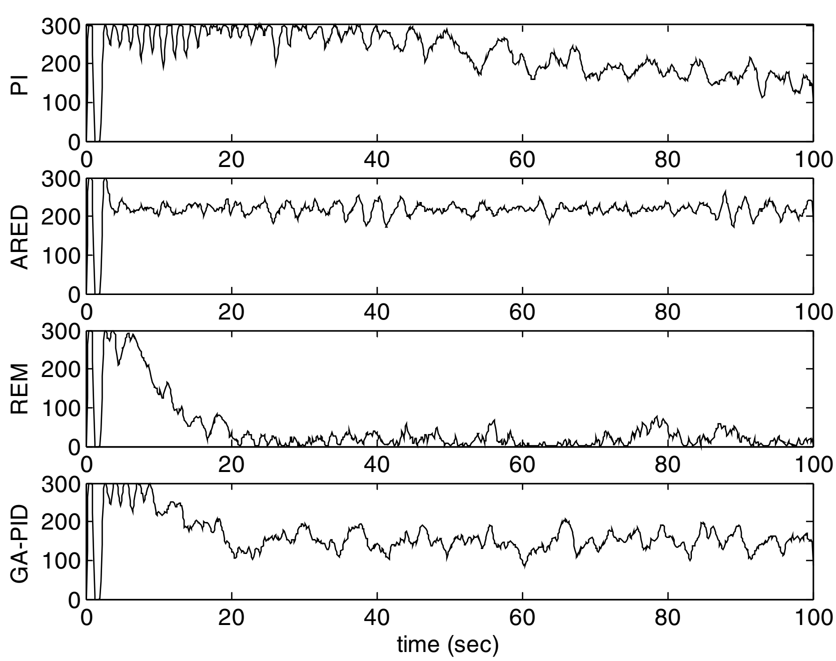

Fig. 5. Responses of queue length (in packets) using PI, ARED, REM, and GA-PID AQM schemes with short propagation delay times.

4.3. Performance of short and long propagation delays

This simulation aims to investigate the robustness of the PI, ARED, REM, and GA-PID AQM schemes against variations of the RTT. We assumed a bandwidth of 10 Mbps and a short inherent propagation delay of 2 ms in the links from the sources to R1 and in the links from R2 to the destinations. There was a bandwidth of 10 Mbps and an inherent propagation delay of 10 ms between R1 and R2. The responses of the queue length obtained with the PI, ARED, REM and GA-PID AQM schemes are shown in Fig. 5. The effect of a long inherent propagation delay was also considered, where the TCP sources and destinations were linked to R1 and R2 with a bandwidth of 10 Mbps and an inherent propagation delay of 20 ms. An inherent propagation delay of 140 ms between R1 and R2 was chosen in this case. The responses of the queue length obtained with the PI, ARED, REM, and GA-PID AQM schemes are shown in Fig. 6. From Figs. 5 and 6, it can be observed that the proposed GA-PID scheme exhibits

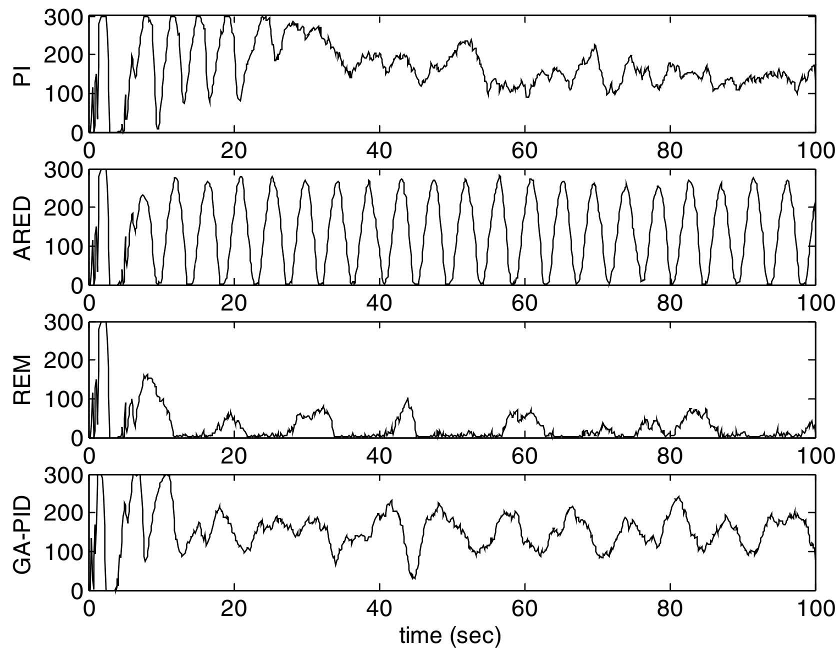

Fig. 6. Responses of queue length (in packets) using PI, ARED, REM, and GA-PID AQM schemes with long propagation delay times.

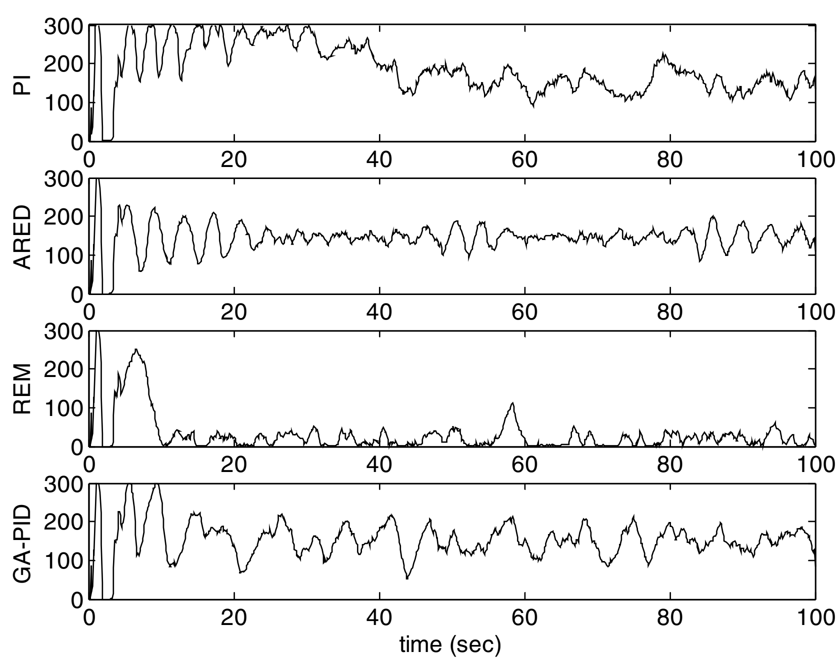

Fig. 7. Responses of queue length (in packets) using PI, ARED, REM, and GA-PID AQM schemes with bursting and unresponsive flows (FTP + UDP).

better performance with respect to queue regulation and expeditious response on different propagation delays. We can obtain that ARED has a steady-state error with a short inherent propagation delay and has serious oscillation when given a long inherent propagation delay. Even though PI has no steady-state error, the transient responses are too sluggish.

4.4. Performance of mixed TCP and UDP connections

Next, to consider more realistic scenarios, bursting and unresponsive flows were introduced into our simulations. In the dumbbell network topology, we set the number of FTP flows to 80 and the number of CBR flows to 50. Each FTP flow was based on TCP-Reno; the CBR flow and bursting ON/OFF flow were based on the User Datagram Protocol (UDP). Each of the UDP sources follows an exponential ON/OFF traffic model, with idle and burst times of a mean of 0 and 0.5 s, respectively, and with a sending rate during on-time of 64 kbps. The FTP flows started at the beginning of the simulation and stopped at the end. The additional bursting CBR flows were active from 20 to 40 s and from 70 to 90 s. The responses of the queue size obtained by using the PI, ARED, REM, and GA-PID AQM schemes are depicted in Fig. 7. The results show that the PI, ARED, and GA-PID scheme can maintain the desired queue length during bursting and unresponsive flows. However, the ARED and GA-PID schemes can react rapidly and regulate the system to achieve the desired queue length.

4.5. Performance of link utilization and packet loss rate

To further evaluate the robustness of the proposed controller, the number of users was varied between 50 and 200, and different propagation delays with diverse times of between 20 and 140 ms, are included in this experiment. The responses of the link utilization and packet loss rate in the different number of users are described in Figs. 8 and 9, respectively. As shown in Fig. 8, we can observe that the utilization of the GA-PID is very close to the PI and ARED schemes in various numbers of TCP connections. Furthermore, the link utilization of PI, ARED and GA-PID are raised when the number of users is increased. In Fig. 9, compared with the PI, ARED and REM schemes, the proposed GA-PID controller has a far better packet loss performance. Although, the utilization of REM is better than others, the packets loss rate is relatively large. Figs. 10 and 11 depict the responses of the packet loss rate and link utilization in the different propagation delays, respectively. Obviously, both link utilization and packet loss rate of the proposed scheme outperforms other AQM schemes.

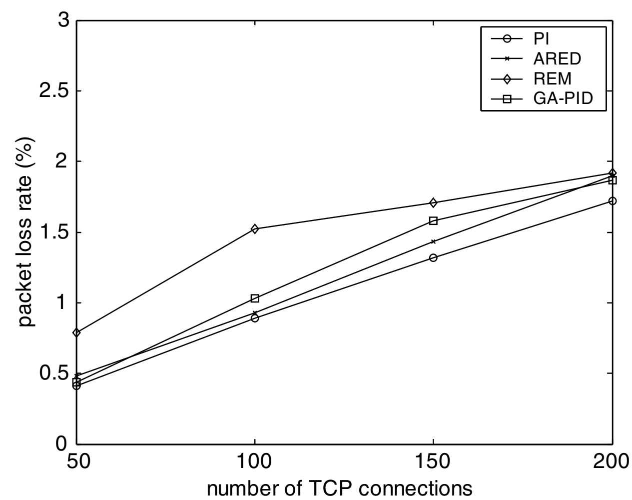

Fig. 9. Packet loss rate vs. number of users for different AQM schemes.

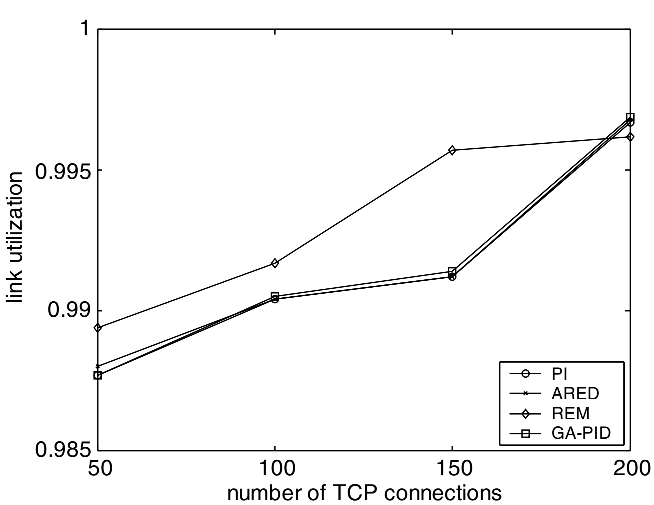

Fig. 8. Link utilization vs. number of users for different AQM schemes.

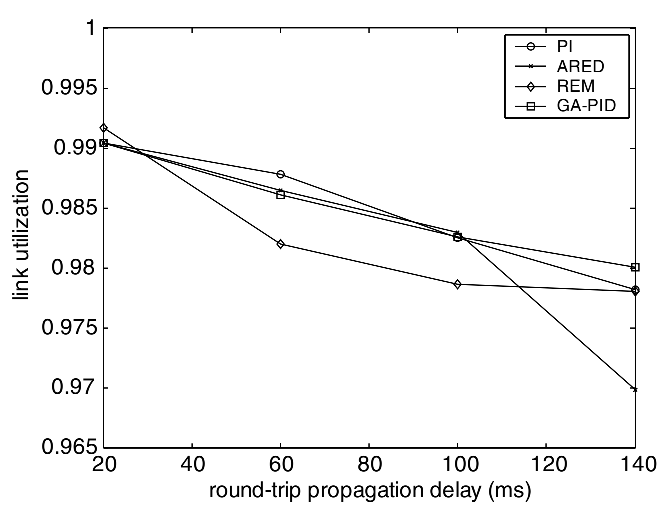

Fig. 10. Link utilization vs. various propagation delays for different AQM schemes.

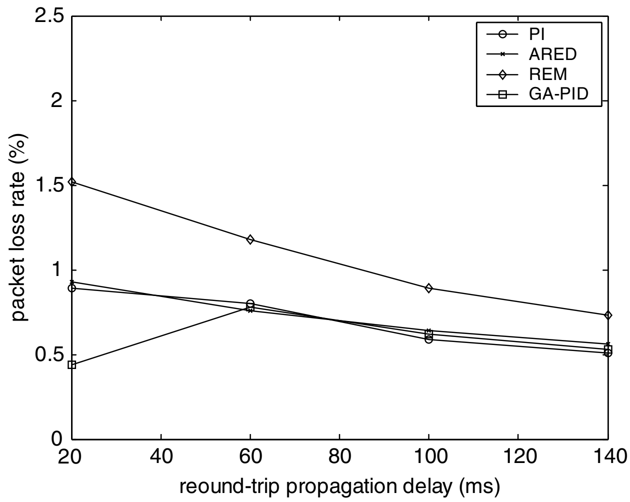

Fig. 11. Packet loss rate vs. various propagation delays for different AQM schemes.

4.6. Performance of multiple links

Finally, we considered a network topology with multiple bottleneck links as shown in Fig. 12. In this case, there were 300 TCP-Reno connections with the senders shown at the left- and right-hand sides, and 50 connections for each cross-traffic sender–receiver pair. The maximum buffer of each router was 200 packets; the bandwidth and propagation delay of each link are also indicated in Fig. 12. Note that link 1–2 and link 5–6 are almost empty, indicating that these two links are not bottleneck links. Another situation can also be found in links 2–3 and 4–5, which operate in a similar fashion. The responses of the queue lengths for link 2–3 and link 3–4 are depicted in Figs. 13 and 14, respectively.

From the above various simulations, it can be evidently observed that the performance and robustness of the proposed GA-PID AQM controller are better than other

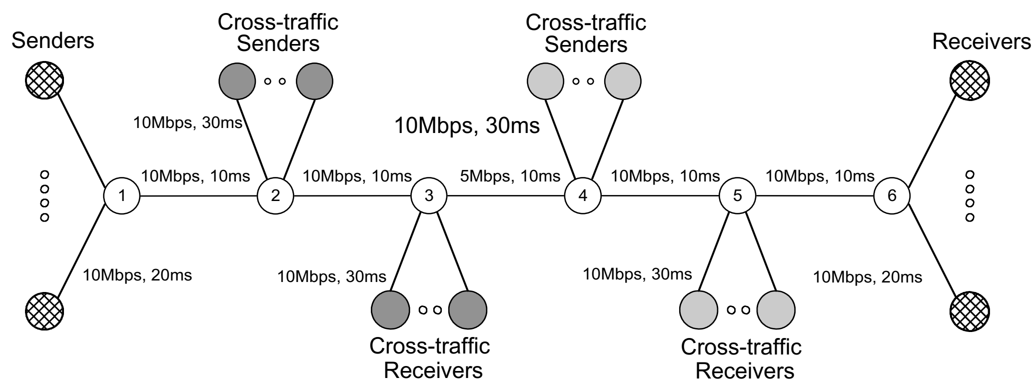

Fig. 12. The network topology with multiple bottleneck links.

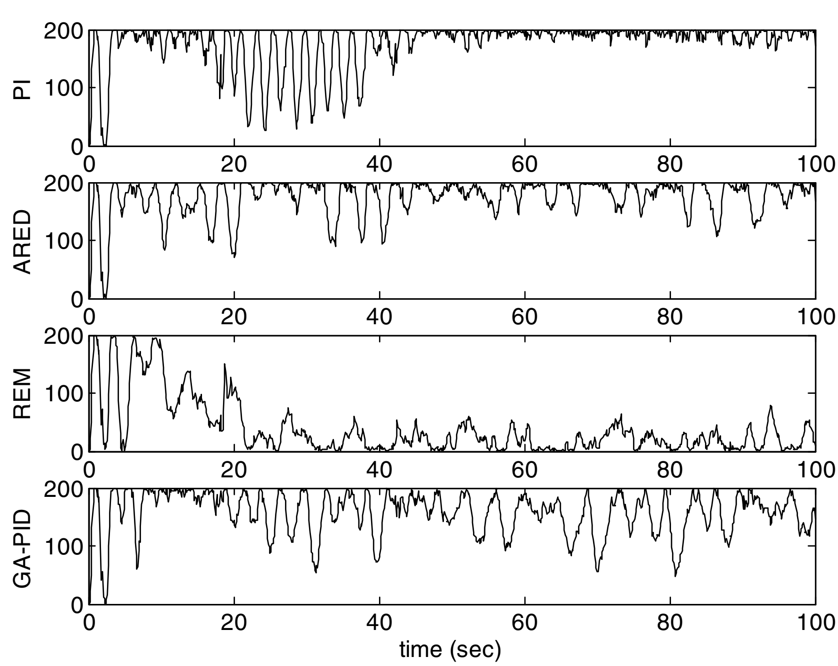

Fig. 13. Evolution of link 2–3 (in packets) using PI, ARED, REM and GA-PID AQM schemes.

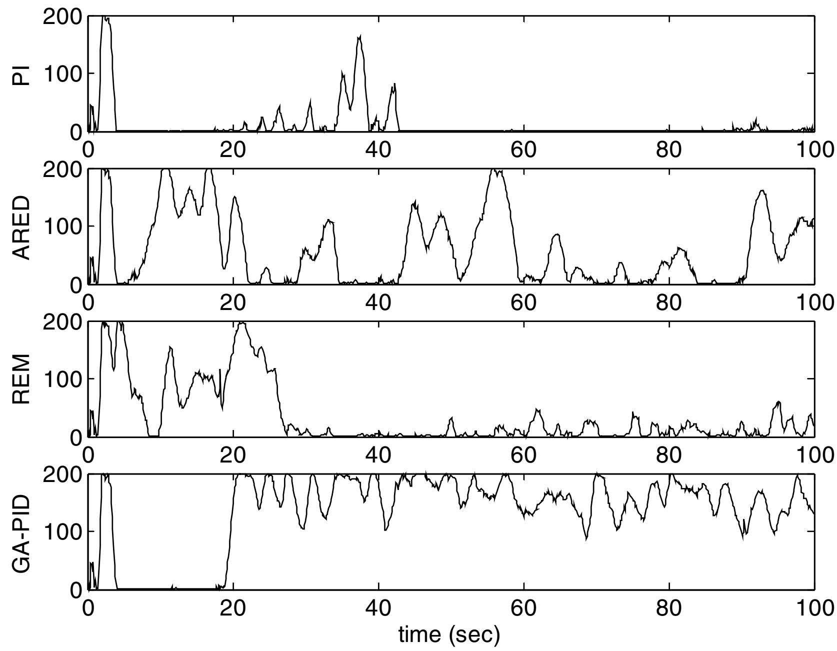

Fig. 14. Evolution of link 3–4 (in packets) using PI, ARED, REM and GA-PID AQM schemes.

AQM controllers and can be applied to more complex network topologies.

5. Conclusions

A GA-based PID controller for AQM has been presented to ensure both high utilization and low packet loss rate by regulating the queue length in an Internet router. Based on the improved genetic algorithm, a simple and effective PID controller has been proposed. Three PID controller gains can be directly obtained by solving the specified optimization problem via genetic algorithm. The performance of the proposed AQM congestion control scheme has been evaluated in various network scenarios via the numerical simulations. The simulation results reveal that the proposed scheme is superior to the existing AQM schemes.

References

Athuraliya, S., Low, S. H., Li, V. H., & Yin, Q. (2001). REM: Active queue management. *IEEE Network Magazine, 15*(3), 48–53.

Bequette, B. W. (2003). *Process control: Modeling, design, and simulation*. New York: Prentice-Hall.

Cao, Y. J. (1997). Eigen value optimization problems via evolutionary programming. *Electronics Letters, 33*(7), 642–643.

Chang, X., & Muppala, J. K. (2006). A stable queue-based adaptive controller for improving AQM performance. *Computer Networks, 50*(13), 2204–2224.

El-Zonkoly, A. M. (2005). Power system model validation for power quality assessment applications using genetic algorithm. *Expert Systems with Applications, 29*(4), 941–944.

Feng, W., Shin, K. G., Kandlur, D. D., & Saha, D. (2002). The BLUE active queue management algorithms. *IEEE/ACM Transactions on Networking, 10*(4), 89–102.

Floyd, S., Gummadi, R., & Shenker, S. (2001). Adaptive RED: An algorithm for increasing the robustness of RED's active queue management. http://www.icir.org/floyd/papers/adaptiveRed.pdf.

Floyd, S., & Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking, 1*(4), 397–413.

Goldberg, D. E. (1989). *Genetic algorithm in search, optimization, and machine learning, reading*. MA: Addison Wesley.

Hollot, C. V., Misra, V., Towsley, D., & Gong, W. B. (2001) On designing improved controllers for AQM routers supporting TCP flows. In *Proceedings of IEEE INFOCOM* (pp. 1726–1734). Alaska.

Huang, C. J., Cheng, C. L., Chuang, Y. T., & Jang, J. S. (2006). Admission control schemes for proportional differentiated services enabled internet servers using machine learning techniques. *Expert Systems with Applications, 31*(3), 458–471.

Johnson, J. M., & Rahmat-Samii, Y. (1997). Genetic algorithms in engineering electromagnetics. *IEEE Antennas and Propagation Magazine, 39*(4), 7–21.

Juang, Y. S., Lin, S. S., & Kao, H. P. (2007). An adaptive scheduling system with genetic algorithms for arranging employee training programs. *Expert Systems with Applications, 33*(3), 642–651.

Kelly, F. (2001). Mathematical modeling of the Internet. In Engquist, B., Schmidt, W. (Eds.), *Mathematics unlimited-2001 and beyond*. (pp. 685–702).

Kim, K. J., Jeong, I. J., Park, J. C., Park, Y. J., Kim, C. G., & Kim, T. H. (2007). The impact of network service performance on customer satisfaction and loyalty: High-speed internet service case in Korea. *Expert Systems with Applications, 32*(3), 822–831.

Li, Y., Ko, K. T., & Chen, G. (2005). A Smith predictor-based PI-controller for active queue management. *IEICE Transactions on Communication, 88*, 4293–4300.

Lin, D., & Morris, R. (1997). Dynamics of random early detection. In *Proceedings of ACM SIGCOM'97* (pp. 127–137). Cannes.

Misra, V., Gong, W. B., & Towsley, D. (2000). Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *Proceedings of ACM/SIGCOM* (pp. 151–160). Stockholm.

Park, E. C., Lim, H., Park, K. J., & Choi, C. H. (2004). Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks. *Computer Networks, 44*(1), 17–41.

Qtt, T. J., Lakshman, T. V., & Wong, L. H. (1999). SRED: Stabilized RED. In *Proceedings of IEEE INFOCOM'99* (pp. 1346–1355). New York.

Ren, F. Y., Lin, C., & Yin, X. H. (2005). Design a congestion controller based on sliding mode variable structure control. *Computer Communications, 28*(9), 1050–1061.

Ren, F., Lin, C., & Wei, B. (2005). A robust active queue management algorithm in large delay network. *Computer Communications, 28*(5), 485–493.

Setnes, M., & Roubos, H. (2000). GA-fuzzy modeling and classification: complexity and performance. *IEEE Transactions on Fuzzy Systems, 8*(5), 509–522.

Srinivas, M., & Patnaik, L. M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithm. *IEEE Transactions on Systems, Man and Cybernetics, 24*(4), 656–667.

Wang, D., & Hollot, C. V. (2003). Robust analysis and design of controllers for a single flow. In *Proceedings of ICCT* (pp. 276–280). Harbin.

Wang, K. J., & Lin, Y. S. (2007). Resource allocation by genetic algorithm with fuzzy inference. *Expert Systems with Applications, 33*(4), 1025–1035.

Wang, X., Wang, Y., Zhou, H., & Huai, X. (2006). PSO-PID: A novel controller for AQM routers. In *Proceedings of IEEE/IFIP WOCN* (pp. 1–5). Bangalore.

Yan, P., Gao, Y., & Özbay, H. (2003). A variable structure control approach to active queue management for TCP with ECN. In *Proceedings of the 8th IEEE symposium on computers and communications* (pp. 1005–1011). Turkey.

Zhang, H. Y., Liu, B. H., & Dou, W. H. (2003). Design of a robust active queue management algorithm based on feedback compensation. In *Proceedings of ACM/SIGCOMM'03* (pp. 277–285). Karlsruhe.