Available online at www.sciencedirect.com
SCIENCE DIRECT®
PERGAMON
Control Engineering Practice 11 (2003) 1127–1142
CONTROL ENGINEERING PRACTICE
www.elsevier.com/locate/conengprac

Anti-windup compensator for active queue management in TCP networks

Kyung-Joon Park[a,*], Hyuk Lim[a], Tamer Başar[b], Chong-Ho Choi[a]

[a] School of Electrical Engineering and Computer Science and ASRI, Seoul National University, Seoul 151-742, South Korea
[b] Coordinated Science Laboratory and Department of Electrical and Computer Engineering, University of Illinois, 1308 West Main Street, Urbana, IL 61801, USA

Received 3 March 2003; accepted 3 March 2003

Abstract

In this paper, we apply a dynamic anti-windup scheme for improving the performance of a conventional proportional–integral (PI) controller for active queue management (AQM) supporting TCP flows. When a PI controller is used for AQM, the *windup* phenomenon of the integral action can cause performance degradation because the packet drop probability is limited between 0 and 1. Therefore we suggest a TCP/AQM model with a saturating actuator and apply a dynamic anti-windup method for improving the performance of the conventional PI AQM scheme. The proposed scheme not only provides graceful performance degradation, but also guarantees the stability of the overall system with the linearized TCP model. We verify the performance of the proposed scheme through *ns-2* simulations. The simulation results show that our scheme outperforms the conventional PI controller when the traffic load is not stationary, which is always the case in real network environment.

Keywords: Active queue management; TCP congestion control; Proportional-integral control; Anti-windup method

1. Introduction

During the past decade, the Internet has been the most influential engineering product. The traffic behavior of the current Internet is mostly governed by TCP dynamics (Stevens, 1994). TCP has been designed for best-effort networks and has been proved to be a great success with the today's Internet. However, current TCP congestion control with drop-tail queues has several problems: First of all, TCP sources of drop-tail queues reduce their rates only after detecting packet loss due to queue overflow. Therefore considerable time may have passed between the packet drop and its detection. At the same time, a large number of packets may be dropped as the sources continue to transmit at a rate that the network cannot support. In addition, packet drop at drop-tail queues could result in the global synchronization of sources (Floyd & Jacobson, 1992).

To alleviate these problems, random early detection (RED) gateways were proposed for active queue management (AQM) (Floyd & Jacobson, 1993). However, the original RED algorithm also has several shortcomings. First, the parameter tuning remains an inexact science (May, Bolot, Diot, & Lyles, 1999; Feng, Kandlur, Saha, & Shin, 1999a, b). Furthermore, there exist some arguments on the deployment of RED, especially in the case of small buffers (May et al., 1999). Many variants of RED have been proposed to resolve these problems (Feng et al., 1999b; Ott, Lakshman, & Wong, 1999; Wang & Shin, 1999; Cnodder, Elloumi, & Pauwels, 2000). Recently, many researchers have proposed system theoretic approaches (Athuraliya, Low, Li, & Yin, 2001; Yin & Low, 2001; Low, Paganini, & Doyle, 2002; Hollot, Misra, Towsley, & Gong, 2001a, b; Kunniyur & Srikant, 2001; Aweya, Ouellette, & Montuno, 2001). The authors Athuraliya et al. (2001), Yin and Low (2001), Low et al. (2002) proposed an optimization-based view of networks and suggested the

*Corresponding author. Tel.: +82-2-880-7313; fax: +82-2-885-4459.
E-mail addresses: kjpark@csl.snu.ac.kr (K.-J. Park), hyuklim@csl.snu.ac.kr (H. Lim), tbasar@decision.csl.uiuc.edu (T. Başar), chchoi@csl.snu.ac.kr (C.-H. Choi).

0967-0661/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0967-0661(03)00059-5

random early marking (REM) AQM algorithm, which is basically a PI controller. In Hollot et al. (2001a, b), the authors gave a control theoretic analysis of RED and designed a PI controller, which outperformed RED significantly. Kunniyur and Srikant (2001) proposed the adaptive virtual queue (AVQ) algorithm, which focused on the rate mismatch instead of the queue size. Aweya et al. (2001) focused on stabilizing the queue and proposed an integral controller for AQM.

In this paper, we are concerned with the performance of the conventional PI controller for AQM. First we model the TCP/AQM system as a system with a saturating actuator because the packet drop probability is naturally constrained to lie between 0 and 1. With this constraint, the output of the integral controller would increase and become large if the queue size remains below the target value over some period. Once this happens, the integral controller cannot regulate the queue size properly as the queue size changes and this could result in a significant performance degradation. This kind of problem is known as the *windup* phenomenon of an integral controller, which occurs when a system consists of an integral controller or a controller with slow dynamics and a saturator at the control input (Peng, Vrancic, & Hanus, 1996; Astrom & Rundqwist, 1989; Kothare, Campo, Morari, & Nett, 1994).

To resolve this problem, we add a saturator to the TCP/AQM model and apply a dynamic anti-windup method to the conventional PI AQM scheme. A static anti-windup compensator was proposed for the PI AQM scheme in Lim, Park, Park, and Choi (2002). However, this static anti-windup compensator only considered the steady state behavior and neglected the delay in control input. Here we adopt a dynamic anti-windup scheme proposed in Park, Chai, and Choo (2000). This scheme is an extension of the anti-windup method in Park and Choi (1995) in the sense that it further takes into account the delay in control input. The objective of the proposed scheme is to provide an additional dynamic compensator that will exhibit graceful performance degradation of the closed-loop system under saturation. We can determine the design parameters of the dynamic compensator explicitly based on the objective function of controller states. The total stability of the overall system is guaranteed if some algebraic conditions hold. We compare the performance of the proposed scheme with the conventional PI controller and the PI controller with the incremental algorithm (Peng et al., 1996) through *ns-2* simulations. The simulation results show that our proposed control scheme outperforms the conventional PI controller when the traffic is fluctuating, which is always the case in the real network environment. Our main contributions in this paper are as follows:

(i) We take into account the limitation that the packet drop probability lies between 0 and 1, and introduce a TCP/AQM model with a saturating actuator.

(ii) To compensate windup phenomenon which results from saturation, we apply a dynamic anti-windup scheme to the conventional PI controller. We show the stability of the overall system in the presence of delay in control input.

The rest of the paper is organized as follows: In Section 2, we present details of the TCP/AQM system with emphasis on the saturating actuator. In Section 3, we describe the proposed anti-windup algorithm. In Section 4, we give the stability analysis of the proposed scheme. In Section 5, we verify the performance of the proposed algorithm via simulation using the *ns-2* network simulator (Fall & Varadhan, 2001). Finally, we present the conclusion in Section 6.

2. Feedback control of active queue management

RED is an AQM algorithm, which controls network congestion by randomly dropping or marking packets with a probability p_d. When TCP sources detect that their packets are dropped or marked, they reduce their sending rates, and the queue size of the router decreases. This process constitutes a closed loop feedback control system as shown in Fig. 1 (Aweya et al., 2001). The system consists of TCP sources, a router queue, and an AQM controller. This block diagram is a logical analysis of the TCP/AQM system and actually the AQM controller is implemented in the router. The congestion controller regulates the queue size of the router by changing the probability p_d. Because the PI controller ensures that the steady state error becomes zero, it is more appropriate than RED for regulation of queue size (Hollot et al., 2001b).

Fig. 2 is the block diagram of the TCP/AQM system with the conventional PI controller. The controller input is the queue size error q_e, which is the difference between the target queue size and the queue size of the router, i.e.

$$q_e(t) = q(t) - q_{ref}, \tag{1}$$

where q_{ref} is the target queue size of the router. The model in Fig. 2 differs from those in Hollot et al. (2001a) and Aweya et al. (2001) in the sense that the restriction on the drop probability is included. The drop probability is obtained by limiting the controller output to the interval $[0, p_{\max}]$ as described by the following equation:

$$p_d(t) = \mathrm{sat}(v(t)), \tag{2}$$

$$\mathrm{sat}(x) = \begin{cases} p_{\max} & x > p_{\max}, \\ 0 & x < 0, \\ x & \text{otherwise}, \end{cases} \tag{3}$$

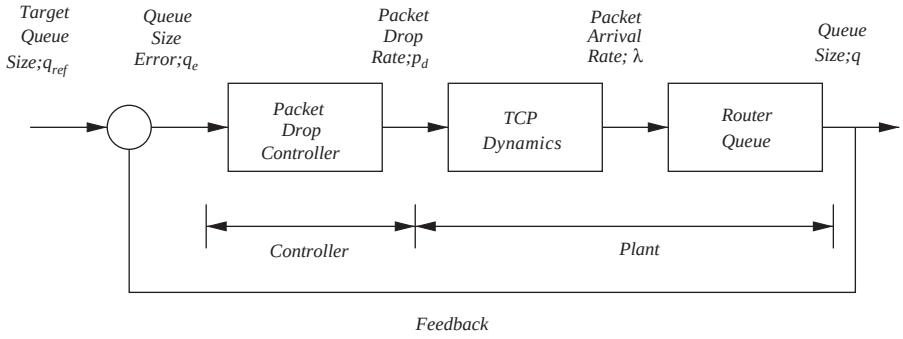

Fig. 1. TCP congestion avoidance as a closed-loop feedback control system.

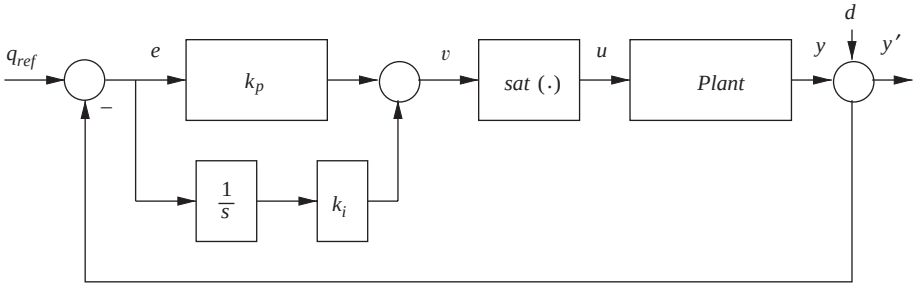

Fig. 2. Conventional PI control.

where $p_{\max}$ is the maximum drop probability. Usually $p_{\max}$ can be set to 1. According to the drop probability, packets are dropped or marked. When the packets are dropped or marked, each of the TCP sources adjusts its window size to reduce its sending rates, and consequently the queue size decreases.

Now note the difficulty that arises from the saturation function. Generally an integral controller or a controller with slow dynamics features *windup* phenomenon (Peng et al., 1996; Astrom & Rundqwist, 1989; Kothare et al., 1994). A PI controller consists of a proportional and an integral controller as follows:

$$v(t) = k_P q_e(t) + k_I \int_0^t q_e(\tau)\,\mathrm{d}\tau, \tag{4}$$

where k_P, k_I are the proportional gain and the integral gain, respectively (Franklin, Powell, & Emami-Naeini, 1995). A PI controller regulates the queue size of the router properly when it operates around the target queue size. However, real network traffic load varies rapidly due to the bursty nature, and sometimes can be much lighter than what is required to maintain the target queue size. If the queue size is smaller than the target queue size for a certain period of time, then the state of the integral controller will become a large negative value, but p_d will remain zero because of the saturation. The rationale for this situation is as follows: A router is not always congested. For a given network topology, as senders and receivers change, congested routers will also be changed. This implies that any router has some period when the incoming rate is smaller than link capacity. During this period, the queue size error, q_e remains negative, which results in a large negative value of integration of q_e. Consequently the controller output, $v(t)$, will be negative. This negative value indicates that the controller tries to increase incoming traffic to match the desired queue size. However it is impossible to increase the incoming rate at the router and the resulting control input, i.e. the drop probability will remain zero as long as $v(t)$ is negative. In summary, during the period of underutilization, $q_e(t)$ is negative and the integral of $q_e(t)$ becomes a large negative value while the drop probability is kept zero. If this situation is followed by rapid traffic increase over the link capacity, buffer overflow will occur because the state of the integral controller will remain negative for a considerable time. In this case, the controller will not work at all and the system will be open-loop, which is virtually a drop-tail queue management. Consequently the performance of the congestion control degrades significantly.

3. Design of an anti-windup compensator

Here we adopt an anti-windup scheme in Park et al. (2000). This scheme is based on the dynamic anti-windup method in Park and Choi (1995), and Park and Choi (1997). However this scheme also takes into account time delay in control input. Here we assume

that the PI controller has already been designed to show satisfactory performance in the absence of saturation. This assumption can be satisfied if we assume that we already have designed a PI controller by following the guideline in Hollot et al. (2001a). Necessary assumptions for design of the proposed anti-windup scheme will be given explicitly later. The objective of the proposed anti-windup scheme is to provide an additional compensator that shows graceful performance degradation of the overall system with saturation. To this end, we reformulate the problem into a state-tracking problem since output is the combination of internal states.

First we formulate the state-space representation of the TCP/AQM system. We use a linearized version of the TCP dynamics in Hollot et al. (2001a). Here we use the TCP rate $\lambda(t)$ instead of the window size $W(t)$ as a state variable. If we consider the delay and the number of connections to be constant for the time being, we have the following TCP model:

$$\dot{\boldsymbol{x}}_p(t) = A\boldsymbol{x}_p(t) + Bu(t-\tau), \tag{5}$$

$$y(t) = C\boldsymbol{x}_p(t) + Du(t-\tau), \tag{6}$$

$$u(t) = \mathrm{sat}(v(t)), \tag{7}$$

where $\boldsymbol{x}_p(t) = (\delta q(t), \delta\lambda(t))^{\mathrm{T}}$, $u(t) = \delta p_d(t)$, $\tau = R$, $A = \binom{01}{0a}$, $B = (0,b)^{\mathrm{T}}$, $C = (1,0)$, $D = 0$, $a = -2N/R^2C_l$, and $b = 2C_l^2/3N$. Here R, N, and C_l are the round-trip time, the number of TCP sessions, and the link capacity, respectively. We only consider the delay in control input $u(t)$ as in Hollot et al. (2001a). Note that b is different from the value in Hollot et al. (2001a). This is because we have changed the proportional constant for b to match the equilibrium. This change of constant was mentioned in several papers, for example Kunniyur and Srikant (2001).

Also we represent the conventional PI controller as follows:

$$\dot{x}_c(t) = Fx_c(t) + Ge(t), \tag{8}$$

$$v(t) = Hx_c(t) + Le(t), \tag{9}$$

$$e(t) = r(t) - y(t), \tag{10}$$

where $F = 0$, $G = 1$, $H = k_I$, $L = k_P$, and $r(t) = q_{ref}$.

The proposed compensation structure is shown in Fig. 3, where a dynamic compensator $M(s,\tau)$ is included, which will be designed appropriately for the overall system to show satisfactory performance under saturation. The dynamics of $M(t,\tau)$ is represented by the following state-space model:

$$\dot{\boldsymbol{x}}_d(t) = \Phi(t,\tau)\boldsymbol{x}_d(t) + \Gamma(t,\tau)\zeta(t), \tag{11}$$

$$\xi(t) = \Sigma(t,\tau)\boldsymbol{x}_d(t) + \Lambda(t,\tau)\zeta(t), \tag{12}$$

$$\zeta(t) = v(t) - u(t). \tag{13}$$

Here the matrices $\Phi(\cdot,\cdot)$, $\Gamma(\cdot,\cdot)$, $\Sigma(\cdot,\cdot)$, and $\Lambda(\cdot,\cdot)$ are design parameters. Now we have the following dynamics of the compensated controller $K_c(s)$:

$$\dot{x}_c(t) = Fx_c(t) + Ge(t) - \xi(t), \tag{14}$$

$$v(t) = Hx_c(t) + Le(t), \tag{15}$$

$$e(t) = r(t) - y(t) - d(t), \tag{16}$$

where $d(t)$ is a disturbance at the plant output. $d(t)$ is responsible for some kind of disturbances such as uncontrolled bursty traffic.

To design the compensator, we need the following assumptions:

(A1) The plant is open-loop stable.

(A2) The controller provides acceptable nominal performance in the absence of the saturating actuator.

These assumptions are standard in designing anti-windup schemes. (A1) can be easily verified from (5)–(7) and (A2) will be also satisfied if we assume that we have designed the PI controller by following the guidelines in Hollot et al. (2001b). With these assumptions, we will design a compensator $M(s,\tau)$ that gives graceful performance degradation with saturation nonlinearity.

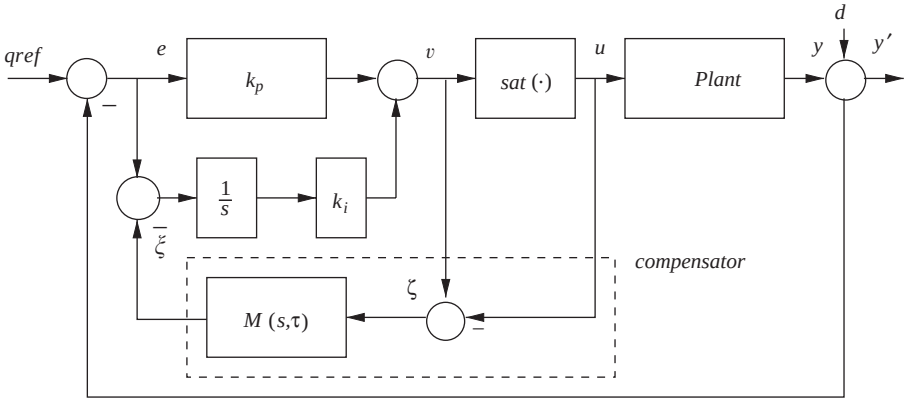

Fig. 3. PI control with the proposed anti-windup compensator.

To design the anti-windup compensator, we need dynamics of the system with and without saturation. In the following subsection, we will derive dynamic equations for the system with and without the saturation, respectively.

3.1. Dynamics of the closed-loop system in the absence of saturation

From (15) and (16) in Park et al. (2000) and by Laplace transform, we obtain the following for the dynamics of the closed-loop TCP/AQM system in the absence of the saturating actuator. This corresponds to the case when $u(t) = v(t)$ in Fig. 3.

$$\begin{bmatrix} sx_c(s) \\ s\boldsymbol{x}_p(s) \end{bmatrix} = A_l(s,\tau)\begin{bmatrix} x_c(s) \\ \boldsymbol{x}_p(s) \end{bmatrix} + B_l(s,\tau)[r(s)-d(s)] + \begin{bmatrix} x_c(0) \\ \boldsymbol{x}_p(0) \end{bmatrix}, \tag{17}$$

where

$$A_l(s,\tau) \coloneqq \begin{bmatrix} P & Q \\ R & S \end{bmatrix}, \qquad B_l(s,\tau) \coloneqq \begin{bmatrix} T \\ U \end{bmatrix}. \tag{18}$$

Here $P = 0$, $Q = -C$, $R = \mathrm{e}^{-s\tau}BH$, $S = A - \mathrm{e}^{-s\tau}BLC$ and $T = G$, $U = \mathrm{e}^{-s\tau}BL$. $(x_c(0), \boldsymbol{x}_p^{\mathrm{T}}(0))^{\mathrm{T}}$ denotes an initial state.

Here we make an additional assumption:

(A3) The system matrix of the closed-loop plant in the absence of saturation, i.e. $A - \mathrm{e}^{-s\tau}BLC$ has stable eigenvalues.

(A3) is necessary for a technical reason and can be easily verified beforehand. Consider the system in Fig. 1 without the saturation, and replace the controller $K(s)$ with L. Then (A3) implies that such a system should be stable. In other words, (A3) requires that the system be stable with only proportional control. (A3) will be satisfied if we assume that the proportional gain k_P has been adjusted by the guideline in Hollot et al. (2001a). Also if we adopt an integral controller only, i.e. $k_P = 0$, then (A3) will be always satisfied with (A2).

3.2. Dynamics of the closed-loop system in the presence of saturation

From (9) and (10) in Park et al. (2000) and (5)–(10), we get the following for the dynamics of the closed-loop TCP/AQM system in the presence of saturation. Here $\tilde{(\cdot)}$ denotes the corresponding state in the presence of saturation.

$$\begin{bmatrix} s\tilde{x}_c(s) \\ s\tilde{\boldsymbol{x}}_d(s) \\ s\tilde{\boldsymbol{x}}_p(s) \end{bmatrix} = A_{sat}(s,\tau)\begin{bmatrix} \tilde{x}_c(s) \\ \tilde{\boldsymbol{x}}_d(s) \\ \tilde{\boldsymbol{x}}_p(s) \end{bmatrix} + B_{sat}(s,\tau) + \begin{bmatrix} \tilde{x}_c(0) \\ \tilde{\boldsymbol{x}}_d(0) \\ \tilde{\boldsymbol{x}}_p(0) \end{bmatrix}, \tag{19}$$

where

$$A_{sat}(s,\tau) \coloneqq \begin{bmatrix} -\Lambda H & -\Sigma & -(G-\Lambda L)C \\ \Gamma H & \Phi & -\Gamma LC \\ 0 & 0 & A \end{bmatrix}, \tag{20}$$

$$B_{sat}(s,\tau) \coloneqq \begin{bmatrix} (G-\Lambda L)(r(s)-d(s)) + \Lambda\widehat{\mathrm{sat}}(\tilde{v}) \\ \Gamma L(r(s)-d(s)) - \Gamma\widehat{\mathrm{sat}}(\tilde{v}) \\ \mathrm{e}^{-s\tau}B\widehat{\mathrm{sat}}(\tilde{v}) \end{bmatrix}. \tag{21}$$

Here $(\tilde{x}_c(0), \tilde{\boldsymbol{x}}_d^{\mathrm{T}}(0), \tilde{\boldsymbol{x}}_p^{\mathrm{T}}(0))^{\mathrm{T}}$ is an initial state and $\mathrm{sat}(\tilde{v})$ denotes the Laplace transform of the controller output due to saturation.

Let $\mu(t,\tau)$ be the impulse response from $\zeta(t)$ to $\xi(t)$. Then we have

$$\mu(t,\tau) \coloneqq \mathscr{L}^{-1}\{M(s,\tau)\} = \mathscr{L}^{-1}\{\Lambda + \Sigma(sI-\Phi)^{-1}\Gamma\}. \tag{22}$$

We can express the controller state as follows:

$$\dot{x}_c(t) = Fx_c(t) + Ge(t) - \mu(t,\tau) * (v(t) - u(t)), \tag{23}$$

where '$*$' denotes the convolution operator. Now we can derive another expression of the system in the presence of the saturating actuator as follows:

$$\begin{bmatrix} s\tilde{x}_c(s) \\ s\tilde{\boldsymbol{x}}_p(s) \end{bmatrix} = A_{aw}(s,\tau)\begin{bmatrix} \tilde{x}_c(s) \\ \tilde{\boldsymbol{x}}_p(s) \end{bmatrix} + B_{aw}(s,\tau) + \begin{bmatrix} \tilde{x}_c(0) \\ \tilde{\boldsymbol{x}}_p(0) \end{bmatrix}, \tag{24}$$

where

$$A_{aw}(s,\tau) \coloneqq \begin{bmatrix} -M(s,\tau)H & -[G - M(s,\tau)L]C \\ 0 & A \end{bmatrix}, \tag{25}$$

$$B_{sat}(s,\tau) \coloneqq \begin{bmatrix} (G - M(s,\tau)L)(r(s)-d(s)) + M(s,\tau)\widehat{\mathrm{sat}}(\tilde{v}) \\ \mathrm{e}^{-s\tau}B\widehat{\mathrm{sat}}(\tilde{v}) \end{bmatrix}. \tag{26}$$

Here we only consider the limitation enforced by the lower bound on the control input, i.e. the saturation when $v(t)$ is below zero. The reason is that, even though the upper bound seems to be hit in our TCP/AQM model in (5)–(7), actually the upper limit is seldom realized and consequently we can ignore the upper bound on control input. This comes from the physical interpretation of the TCP/AQM system. When the control input, $u(t)$ is 1, this corresponds to the case of

dropping all the incoming packets. Hence, the effective incoming traffic load will be zero, and the queue size will be decreased immediately. This physical interpretation is not apparent if we only work with the model equations (5)–(7). With these equations, it would be reasonable to consider the upper limit of the saturation. These discrepancies result from the TCP model we use. The model is not valid if the average value of the drop probability $u(t)$ is quite large for some time period. With a large value of the drop probability, TCP behavior is not in a stable regime at all, and the assumption on the average TCP behavior to be in a linear increasing regime will be violated (Mathis, Semke, & Mahdavi, 1997). Most of the window behavior will be in a transient regime. This is why we gave a physical interpretation for upper saturation case.

The controller state with saturation $\tilde{x}_c(t)$ is quite different from $x_c(t)$ since the feedback loop is virtually broken with saturation. Our approach here is to make the states of a saturated system as close as possible to the states of an unsaturated system. We introduce the following performance index:

$$\min_{M(s,\tau)} J = \int_0^\infty \|x_c(t) - \tilde{x}_c(t)\|^2 \, \mathrm{d}t, \tag{27}$$

where $\|\cdot\|$ denote the *Euclidean norm*.

First we state the following lemma.

Lemma 1. *Under Assumptions* (A1)–(A3), *the dynamic compensator* $M^*(s,\tau)$ *which minimizes the performance index* J *is uniquely determined by*

$$\Phi^*(s,\tau) = A - \mathrm{e}^{-s\tau}BLC, \tag{28}$$

$$\Gamma^*(s,\tau) = \mathrm{e}^{-s\tau}B, \tag{29}$$

$$\Sigma^*(s,\tau) = GC, \tag{30}$$

$$\Lambda^*(s,\tau) = 0. \tag{31}$$

Proof. Since this lemma is a slight modification of Lemma 3.1 in Park et al. (2000), we will closely follow the proof of Lemma 3.1 in Park et al. (2000). By Parseval's theorem, the performance index J can be expressed as

$$J = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \|x_c(s) - \tilde{x}_c(s)\|^2 \, \mathrm{d}s. \tag{32}$$

The matrix $(sI - A_l(s,\tau))^{-1}$ exists since $A_l(s,\tau)$ is stable from Assumption (A2). Also $(sI - S)$ exists by (A3). From the inversion lemma of block matrices, we get

$$(sI - A_l(s,\tau))^{-1} = \begin{bmatrix} -X(s) & -X(s)Q(sI-S)^{-1} \\ -(sI-S)^{-1}RX(s) & -W(s) \end{bmatrix}, \tag{33}$$

where

$$X(s) := -[(sI - P) - Q(sI - S)^{-1}R]^{-1},$$
$$W(s) := -(sI - S)^{-1} + (sI - S)^{-1}RX(s)Q(sI - S)^{-1}.$$

From (17) and (33), we get

$$x_c(s) = -X(s)(T + Q(sI - S)^{-1}U)[r(s) - d(s)] - X(s)x_c(0) - X(s)Q(sI - S)^{-1}\boldsymbol{x}_p(0). \tag{34}$$

The system matrix A is stable by (A1). Here, tentatively assume that $sI + M(s,\tau)H$ is nonsingular for all $Re[s] \geqslant 0$. Under this assumption, $(sI - A_{aw}(s,\tau))^{-1}$ exists. Now, first we find $M^*(s,\tau)$ that minimizes J, and then we will show that $sI + M^*(s,\tau)H$ is nonsingular. By applying the inversion lemma of block matrices, we have

$$[sI - A_{aw}(s,\tau)]^{-1} = \begin{bmatrix} [sI + M(s,\tau)H]^{-1} & -[sI + M(s,\tau)H]^{-1}[G - M(s,\tau)L] \\ 0 & C(sI - A)^{-1} \\ & (sI - A)^{-1} \end{bmatrix}. \tag{35}$$

From (24) and (35), we get

$$\begin{aligned} \tilde{x}_c(s) = {} & [sI - M(s,\tau)H]^{-1}(G - M(s,\tau)L[r(s) - d(s)] \\ & + [sI + M(s,\tau)H]^{-1}M(s,\tau)\widehat{\mathrm{sat}}(\tilde{v}) \\ & - \mathrm{e}^{-s\tau}[sI + M(s,\tau)]^{-1}[G - M(s,\tau)L] \\ & \times C(sI - A)^{-1}B\,\widehat{\mathrm{sat}}(\tilde{v}) \\ & + [sI + M(s,\tau)H]^{-1}\tilde{x}_c(0) \\ & - [sI + M(s,\tau)H]^{-1}[G - M(s,\tau)L] \\ & \times C(sI - A)^{-1}\tilde{\boldsymbol{x}}_p(0). \end{aligned} \tag{36}$$

From (34) and (36) together with $(x_c(0), \boldsymbol{x}_p^{\mathrm{T}}(0))^{\mathrm{T}} = (\tilde{x}_c(0), \tilde{\boldsymbol{x}}_p^{\mathrm{T}}(0))^{\mathrm{T}}$,

$$x_c(s) - \tilde{x}_c(s) = \Xi_1(s)[r(s) - d(s)] + \Xi_2(s)\widehat{\mathrm{sat}}(\tilde{v}) + \Xi_3(s)x_c(0) + \Xi_4(s)\boldsymbol{x}_p(0), \tag{37}$$

where

$$\Xi_1(s) := -X(s)(T + Q(sI - S)^{-1}U) - [sI + M(s,\tau)H]^{-1}(G - M(s,\tau)L),$$

$$\Xi_2(s) := [sI + M(s,\tau)]^{-1}[\mathrm{e}^{-s\tau}(G - M(s,\tau)L) \times C(sI - A)^{-1}B - M(s,\tau)],$$

$$\Xi_3(s) := -X(s) - [sI + M(s,\tau)]^{-1},$$

$$\Xi_4(s) := -X(s)Q(sI - S)^{-1} + [sI + M(s,\tau)]^{-1}[G - M(s,\tau)L]C(sI - A)^{-1}.$$

Let $M_t(s,\tau)$ be a solution for $\Xi_2(s)=0$, then

$$M_t(s,\tau)=\mathrm{e}^{-s\tau}GC(sI-A)^{-1} \times B[I+\mathrm{e}^{-s\tau}LC(sI-A)^{-1}B]^{-1}. \quad (38)$$

By applying the matrix inversion lemma, we can express (38) as follows:

$$M_t(s,\tau)=\mathrm{e}^{-s\tau}GC[sI-A+\mathrm{e}^{s\tau}BLC]^{-1}B. \quad (39)$$

By using (27), we get $\Xi_1(s)=0$, $\Xi_3(s)=0$, and $\Xi_4(s)=0$. Consequently, $M_t(s,\tau)$ is an optimal solution $M^*(s,\tau)$ that minimizes J with $J=0$. By comparing (11) and (12) with (39), we get

$$\Phi^*(s,\tau)=A-\mathrm{e}^{-s\tau}BLC, \quad (40)$$

$$\Gamma^*(s,\tau)=\mathrm{e}^{-s\tau}B, \quad (41)$$

$$\Sigma^*(s,\tau)=GC, \quad (42)$$

$$\Lambda^*(s,\tau)=0. \quad (43)$$

Now we show that $sI+M^*(s,\tau)H$ is nonsingular for all $Re[s]\geqslant 0$. From (A2), $sI-A_l(s,\tau)$ is nonsingular for all $Re[s]\geqslant 0$. Thus, we have

$$\det(sI-A_l(s,\tau)) = \det\begin{bmatrix} sI-P & -Q \\ -R & sI-S \end{bmatrix}, \quad (44)$$

$$= \det(sI-S)\det[(sI-P)-Q(sI-S)^{-1}R] \quad (45)$$

$$= \det(sI-S)\det[SI+M^*(s,\tau)H] \quad (46)$$

$$\neq 0, \quad \forall Re[s]\geqslant 0. \quad (47)$$

Therefore, $\det[sI+M^*(s,\tau)H]\neq 0$, $\forall Re[s]\geqslant 0$. This completes the proof. □

From Lemma 1, we have the following theorem for design of the compensator.

Theorem 1. *Consider the dynamic anti-windup scheme in Fig. 3. If Assumptions* (A1)–(A3) *are satisfied, then the state-space realizable form of the dynamic compensator* $M^*(s,\tau)$ *which minimizes the performance index* J *in* (27) *is uniquely determined by the following equations*:

$$\dot{\boldsymbol{x}}_d(t)=A\boldsymbol{x}_d(t)+B\eta(t), \quad (48)$$

$$\xi(t)=C\boldsymbol{x}_d(t), \quad (49)$$

$$\eta(t)=\zeta(t-\tau)-L\xi(t-\tau) \quad (50)$$

and $J=0$, i.e. $\tilde{x}_c(t)=x_c(t)$, $\forall t\in[0,\infty]$ *with* $M^*(s,\tau)$.

Proof. From Lemma 1, we have the following parameters for $M^*(s,\tau)$:

$$\Phi^*(s,\tau)=A-\mathrm{e}^{-s\tau}BLC, \quad (51)$$

$$\Gamma^*(s,\tau)=\mathrm{e}^{-s\tau}B, \quad (52)$$

$$\Sigma^*(s,\tau)=GC, \quad (53)$$

$$\Lambda^*(s,\tau)=0. \quad (54)$$

However, the parameters Φ^*, Γ^*, Σ^*, and Λ^* are not directly realizable forms. For the realization of the dynamic compensator, we need a direct expression of $M^*(s,\tau)$. By substituting (28)–(31) with the zero initial states for Φ, Γ, Σ, and Λ in (11) and (12), we get the following equations:

$$s\boldsymbol{x}_d(s)=(A-\mathrm{e}^{-s\tau}BLC)\boldsymbol{x}_d(s)+\mathrm{e}^{-s\tau}B\zeta(s), \quad (55)$$

$$\xi(s)=GC\boldsymbol{x}_d(s). \quad (56)$$

Equations (55) and (56) can be transformed into

$$s\boldsymbol{x}_d(s)=A\boldsymbol{x}_d(s)+B\eta(s), \quad (57)$$

$$\xi(s)=C\boldsymbol{x}_d(s), \quad (58)$$

where

$$\eta(s)\coloneqq -\mathrm{e}^{-s\tau}LC\boldsymbol{x}_d(s)+\mathrm{e}^{-s\tau}\zeta(s). \quad (59)$$

Furthermore, (59) can be rearranged as

$$\eta(s)=-\mathrm{e}^{-s\tau}LC\boldsymbol{x}_d(s)+\mathrm{e}^{-s\tau}\zeta(s) \quad (60)$$

$$=-\mathrm{e}^{-s\tau}(-LC\boldsymbol{x}_d(s)+\zeta(s)) \quad (61)$$

$$=-\mathrm{e}^{-s\tau}(\zeta(s)-L\xi(s)). \quad (62)$$

The time-domain representation of (62) is

$$\eta(t)=\zeta(t-\tau)-L\xi(t-\tau). \quad (63)$$

Consequently, if we consider the time-domain representation of (57), (58), and (63) together, we have the following equations for a realizable form of the compensator.

$$\dot{\boldsymbol{x}}_d(t)=A\boldsymbol{x}_d(t)+B\eta(t), \quad (64)$$

$$\xi(t)=C\boldsymbol{x}_d(t), \quad (65)$$

$$\eta(t)=\zeta(t-\tau)-L\xi(t-\tau). \quad (66)$$

Also, from Lemma 1, we have $\tilde{x}_c(t)=x_c(t)$, $\forall t\in[0,\infty]$ with $M^*(s,\tau)$. This completes the proof. □

Remark 1. The purpose of the proposed anti-windup scheme is to maintain the controller states to be exactly the same as those without saturation during the period of saturation. However, this does not mean that the controller with saturation is equivalent to that without saturation. What is done in the proposed anti-windup scheme is to match the controller state. Still the matrix L in (9) reflects the saturation effect. If we use a pure

integral controller as in Aweya et al. (2001), then $L = 0$ and the compensated controller with saturation will be the same as that without saturation.

Remark 2. We can design the proposed anti-windup compensator using only some given nominal model of the system. The system parameters are the round trip time R and the number of connections N. For given nominal values of these parameters, i.e. $R = R_0$ and $N = N_0$, we can design the corresponding compensator by Theorem 1. However, in a real network environment, these parameters are always time-varying. In this situation, the equality $\tilde{x}_c(t) = x_c(t), \forall t \in [0, \infty]$ will no longer be valid. Still, $\tilde{x}_c(t)$ with $M^*(s, \tau)$ will follow the controller state of the nominal unsaturated system. Consequently, if we assume that $R \leqslant \bar{R}$ and $N \geqslant \underline{N}$ for some $\bar{R}$ and $\underline{N}$ and design the compensator based on these $\bar{R}$ and $\underline{N}$, then, since the PI controller has been designed beforehand to show good performance with the nominal model, the proposed compensator will be expected to show satisfactory behavior regardless of some model uncertainties. We will give an additional remark on the stability of the overall system with model uncertainties in the next section.

Remark 3. The main purpose of the anti-windup scheme is to compensate the controller when the difference between $v(t)$ and $u(t)$ is observed. However, if the proposed compensator is added to the controller, it can affect the controller for a considerable period even after the actuator escapes from saturation. Consequently, this can affect the controller adversely. Even though the controller states are maintained to the unsaturated controller states by the compensator, this may not be the best way for performance when the actuator escapes from saturation. Hence, when the actuator escapes from saturation, it is desirable to turn off the compensator. This can be achieved by resetting the dynamic compensator states to be zero. This resetting process does not alter the form of the dynamic compensator (Park & Choi, 1995).

4. Stability of the proposed scheme

In this section, we consider stability of the TCP/AQM system with the proposed anti-windup scheme. Here, we establish the stability of the system based on the linearized TCP model (3)–(5). The state representations of closed-loop systems with and without saturation are given in (17) and (19), respectively. We first consider the *total stability* (Chen, 1999) of the overall system. A system is *totally stable* if closed-loop transfer function of every possible input–output pair is bounded-input bounded-output (BIBO) stable. Consequently, total stability represents the bounded-input bounded-state (BIBS) stability. Though the system is open-loop stable, stability of the controller is not guaranteed. Since the PI controller may be unstable, it is necessary to check that all the states of the plant as well as the controller are bounded.

Theorem 2. *The overall system is totally stable under Assumptions* (A1)–(A3).

Proof. Here, we closely follow the proof of Theorem 5.1 in Park et al. (2000). First, note that the compensator itself is stable since Φ^* in (28), the overall closed-loop system matrix without saturation is stable by (A3). Now by substituting (28)–(31) into (20), we get

$$A_{sat}(s,\tau) = \begin{bmatrix} A_l(s,\tau) & -C \\ & -\mathrm{e}^{s\tau}BLC \\ 0 \quad 0 & A \end{bmatrix}, \tag{67}$$

where $A_l(s, \tau)$ is given in (18). The eigenvalues of $A_{sat}(s, \tau)$ consist of the eigenvalues of $A_l(s, \tau)$ and A. Thus, $A_{sat}(s, \tau)$ is stable from (A1) and (A2). Consequently, the theorem follows directly from the fact that (19) is stable since $A_{sat}(s, \tau)$ is stable and sat$(\cdot)$ is a bounded function. □

When the windup phenomenon occurs, the feedback loop will be virtually broken and the system will show open-loop behavior. Consequently if the controller is not stable, the controller state can become very large in the case of windup. With the proposed compensator, the boundedness of the controller state will be guaranteed from Theorem 2.

Now we address a sufficient condition for asymptotic stability. Here we assume that $R \leqslant \bar{R}$ and $N \geqslant \underline{N}$ for some known $\bar{R}$ and $\underline{N}$ and the anti-windup compensator has been designed based on these $\bar{R}$ and $\underline{N}$.

Theorem 3. *The overall system is* asymptotically stable *in the absence of exogenous inputs* $r(t)$ *and* $d(t)$ *if the following condition holds*:

$$k_P < \frac{6\underline{N}^3}{(\bar{R}C_l)^4}. \tag{68}$$

Proof. From the *circle criterion* of *Theorem 5.3* in Park et al. (2000), the system is asymptotically stable if $Re\{\mathrm{e}^{-\mathrm{j}\omega\tau}LP(\mathrm{j}\omega)\} > -1, \forall \omega \in [-\infty, \infty]$. From (5) and (6), we have the following transfer function of the plant:

$$P(\mathrm{j}\omega) = \frac{b}{\mathrm{j}\omega(\mathrm{j}\omega - a)}, \tag{69}$$

where $a = -2\underline{N}/\bar{R}^2C_l$, and $b = 2C_l^2/3\underline{N}$.

From (69) together with $L = k_P$ and $\tau = \bar{R}$, we have the following condition:

$$Re\left\{\frac{b\mathrm{e}^{-\mathrm{j}\omega\bar{R}}}{\mathrm{j}\omega(\mathrm{j}\omega - a)}\right\}k_P > -1, \quad \forall\omega\in[-\infty, \infty]. \tag{70}$$

Note that

$$Re\left\{\frac{b\mathrm{e}^{-\mathrm{j}\omega\bar{R}}}{\mathrm{j}\omega(\mathrm{j}\omega - a)}\right\} = \frac{-b\omega\cos(\omega\bar{R}) + ab\sin(\omega\bar{R})}{\omega(\omega^2 + a^2)}. \tag{71}$$

Since (71) is an even function of ω, we only consider $\omega\in(0, \infty]$. The right hand side of (71) can be bounded by the following inequality:

$$\frac{-b\omega\cos(\omega\bar{R}) + ab\sin(\omega\bar{R})}{\omega(\omega^2 + a^2)} > \frac{b(-\omega + a)}{\omega(\omega^2 + a^2)}. \tag{72}$$

Here, if we let $g(\omega) \coloneqq (-\omega + a)/(\omega(\omega^2 + a^2))$, then $g'(\omega) > 0$ for $\forall\omega > 0$. Therefore

$$g(\omega) > \inf_{\omega\in(0,\infty]} g(\omega) = \lim_{w\to 0^+} g(\omega) = -\frac{1}{a^2}. \tag{73}$$

Consequently,

$$k_P < \frac{a^2}{b} \tag{74}$$

$$= \frac{6\underline{N}^3}{(\bar{R}C_l)^4}. \qquad \square \tag{75}$$

Remark 4. As mentioned in Remark 5.2 in Park et al. (2000), we can readily show that Theorems 2 and 3 are still valid with model uncertainties if the following three conditions hold: (i) The TCP/AQM system is open-loop stable under uncertainties. (ii) The conventional PI controller has been designed beforehand to show robust stability and performance. (iii) The proposed anti-windup compensator $M^*(s, \tau)$ is designed for the nominal model. Since the TCP/AQM is open-loop stable with any values of R and N as we can see from (5)–(7), the first condition is satisfied. Also, if we assume that $R \leqslant \bar{R}$ and $N \geqslant \underline{N}$ for some known $\bar{R}$ and $\underline{N}$, then we can design the PI controller to show robust stability and performance under $R \leqslant \bar{R}$ and $N \geqslant \underline{N}$ by following the guidelines in Hollot et al. (2001b). Consequently, the second condition can also be satisfied. The third condition holds if we design the compensator $M^*(s, \tau)$ with $\bar{R}$ and $\underline{N}$. In summary, we can guarantee the total stability and the asymptotic stability without exogenous inputs regardless of model uncertainties. In a practical sense, however, it may be problematic how to estimate $\bar{R}$ and $\underline{N}$ at routers. We may get a rough estimate of $\bar{R}$ by considering the maximum propagation delay, the buffer size of the router, and the maximum number of hops. Also we can get information on $\underline{N}$ by observing the Internet traffic. Since we do not need to know exact values of R and N, but the rough bound of R and N, it would not be very difficult to estimate these values.

5. Simulations

For a digital implementation, we need to convert the differential equations in Theorem 1 into difference equations. First we rewrite the compensator equations here with $\boldsymbol{x}_d(t) \coloneqq (x_{d1}(t), x_{d2}(t))^{\mathrm{T}}$.

$$\begin{bmatrix}\dot{x}_{d1}(t)\\ \dot{x}_{d2}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & a\end{bmatrix}\begin{bmatrix}x_{d1}(t)\\ x_{d2}(t)\end{bmatrix} + \begin{bmatrix}0\\ b\end{bmatrix}\eta(t), \tag{76}$$

$$\xi(t) = x_{d1}(t), \tag{77}$$

$$\eta(t) = \zeta(t - \tau) - k_P\xi(t - \tau), \tag{78}$$

where $a = -2N/R^2C_l$, and $b = 2C_l^2/3N$.

For design of the compensator, we assume that the minimum number of connections $\underline{N}$ and the maximum round trip time $\bar{R}$ are known. Then, based on these nominal values together with a given value of the link capacity C_l, we design a proposed anti-windup compensator.

Now we have the following equations:

$$\dot{x}_{d1}(t) = x_{d2}(t), \tag{79}$$

$$\dot{x}_{d2}(t) = -\frac{2\underline{N}}{\bar{R}^2C_l}x_{d2}(t) + \frac{2C_l^2}{3\underline{N}}\eta(t), \tag{80}$$

$$\xi(t) = x_{d1}(t), \tag{81}$$

$$\eta(t) = \zeta(t - \bar{R}) - k_P\xi(t - \bar{R}). \tag{82}$$

These can be converted into difference equations with an additional assumption that the maximum delay $\bar{R}$ is a multiple of the sampling time T_s, i.e. $\bar{R} = mT_s$. Several simple methods can be applied to get difference equations such as the forward rectangular rule (also known as Euler's rule), the backward rectangular rule, and the trapezoid rule (also called the bilinear transformation), etc. (Franklin, Powell, & Workman, 1990). Here we adopted Euler's rule for the approximation as follows:

$$x_{d1}((n + 1)T_s) = x_{d1}(nT_s) + T_sx_{d2}(nT_s), \tag{83}$$

$$x_{d2}((n + 1)T_s) = \left(1 - T_s\frac{2\underline{N}}{\bar{R}^2C_l}\right)x_{d2}(nT_s) + T_s\frac{2C_l^2}{3\underline{N}}\eta(nT_s), \tag{84}$$

$$\xi(nT_s) = x_{d1}(nT_s), \tag{85}$$

$$\eta(nT_s) = \zeta((n - m)T_s) - k_P\xi((n - m)T_s). \tag{86}$$

In all simulations, the following nominal values were used for design of the proposed anti-windup compensator: $\underline{N} = 100$ connections, $\bar{R} = 50$ ms, $T_s = 10$ ms, $\bar{R}/T_s = 5$, $k_P = 0.0015$, $k_I = 0.001$, and $q_{ref} = 50$ packets.

We compare the performance of the proposed scheme with the conventional PI control AQM scheme in Hollot et al. (2001b) and the incremental algorithm (Peng et al., 1996). Note that, even though the authors Hollot et al. (2001b) did not consider the windup phenomenon in their work, they implemented a simple and intuitive code to prevent windup phenomenon, which is included in the daily snapshot of *ns-2*. However, it is a heuristic approach for resolving windup problems. In the following simulations, instead of using the PI controller implemented in *ns-2* daily snapshot, we use a PI controller which does not adopt any anti-windup scheme for comparison with our proposed scheme.

For Experiments 1–3, we implemented a dumbbell topology where all connections traverse a single bottleneck link. All other links have the same bandwidth so that they do not create any bottlenecks. The allocated buffer size of the bottleneck router is set to 100 packets. The link capacity is 15 Mb/s and the average packet size is 1000 Bytes.

In Experiment 4, we have considered a multiple-node case as shown in Fig. 9. There are three queues in the network and Queue 2 is shared with the main and cross traffics. We observed Queue 2 under dynamic connection establishment and termination.

5.1. Experiment 1

In this experiment, we verify the performance of the algorithms for three traffic cases. In the first case, we set the real traffic same as the nominal model, i.e. we let $R = 50$ ms and $N = 100$. So the model used for design of the proposed compensator is valid in this case. All the TCP connections are abruptly established and disconnected at $t = 5$ and 30 s, respectively. In the second interval, the number of connections abruptly increases and decreases as follows: $N = (3 - |k - 2|)/2$ during the interval of $t = [40 + 5k, 45 + 5k]$, $k = 0, 1, 2, 3, 4$. Finally we randomly increases and decreases connections during $t = [75, 80]$ and $[95, 100]$, respectively. Fig. 4 shows the result of the conventional PI control. The queue size error during the first under-utilization period makes the output of the PI controller become a large negative value as we can see from Fig. 4. This negative value is maintained till about $t = 10$ s. Hence, after the connections are established at $t = 5$ s, during the period of $t = [5, 10]$, the PI controller does not work any more, and the queue becomes a drop-tail, which results in severe oscillations as we can see from Fig. 4. Hence, the performance degrades significantly. This windup phenomenon is observed in Fig. 4 when sudden connections are established. During the second interval, the windup phenomenon is more severe because the under-utilization periods are longer than the first interval. The queue size is not properly controlled and decreased until around $t = 60$ s. We can also see a similar windup

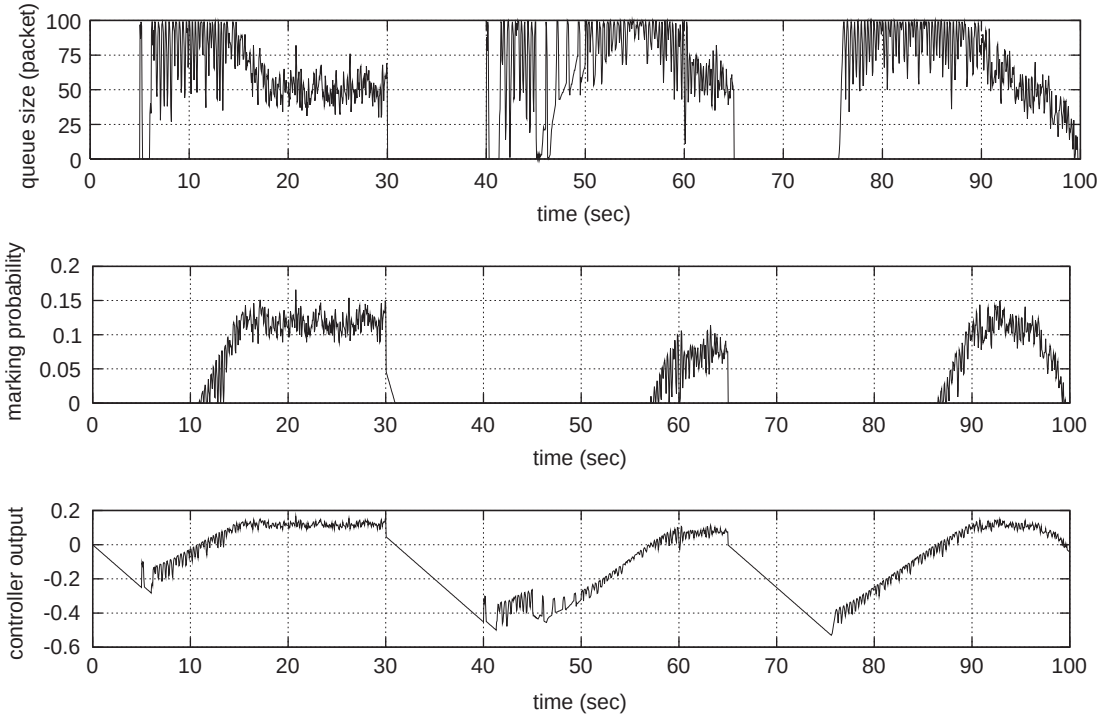

Fig. 4. Experiment 1; Conventional PI control in a single bottleneck topology.

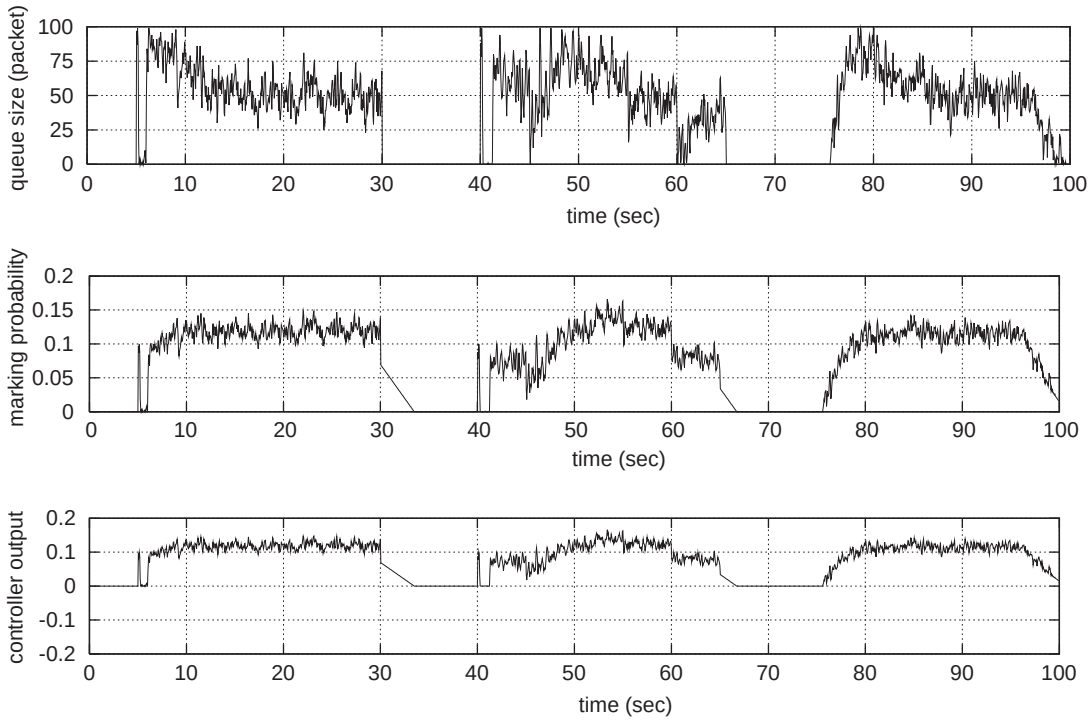

Fig. 5. Experiment 1; PI control with the incremental algorithm in a single bottleneck topology.

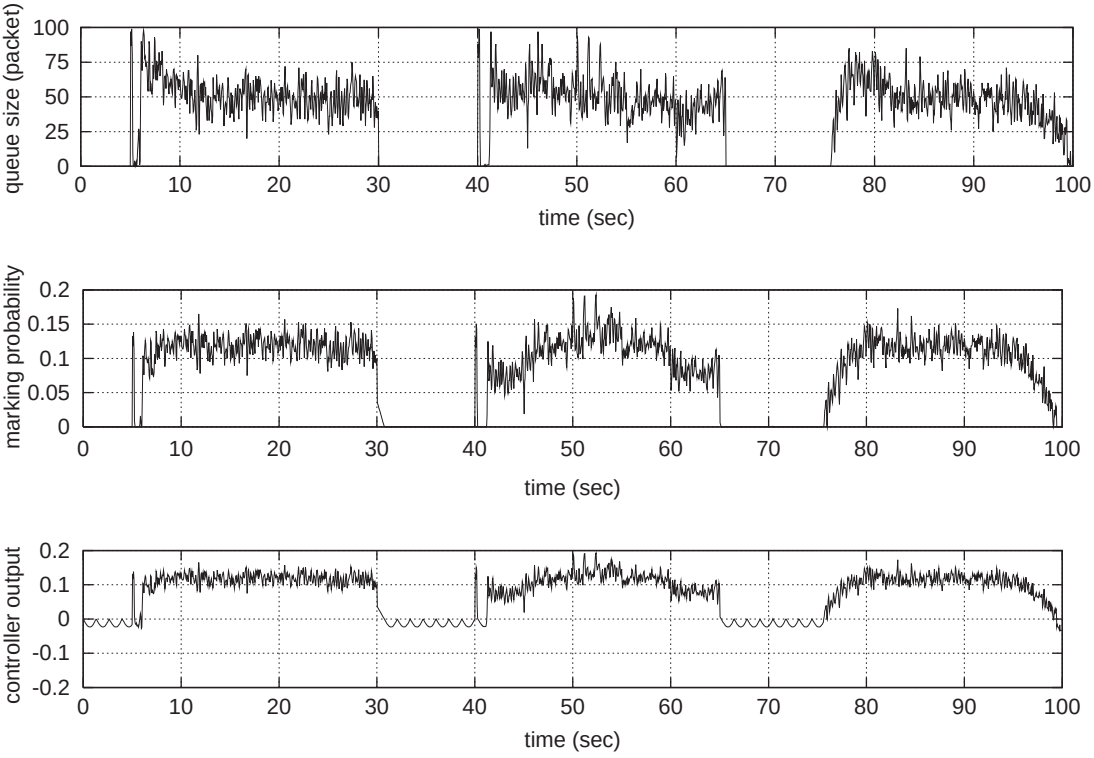

Fig. 6. Experiment 1; PI control with the proposed scheme in a single bottleneck topology.

phenomenon in the third interval. Figs. 5 and 6 show the simulation results of the incremental algorithm and the proposed scheme, respectively. In both cases, the controller output does not become a large negative value and remains around zero during the under-utilization period. However, from Fig. 5, we can know that

the incremental algorithm does not control the queue properly at around $t = 45$ and 60 s, which results from the slow response of the incremental algorithm.

5.2. Experiment 2

Now we perform the same scenario with different values of R and N. The purpose of this experiment is to

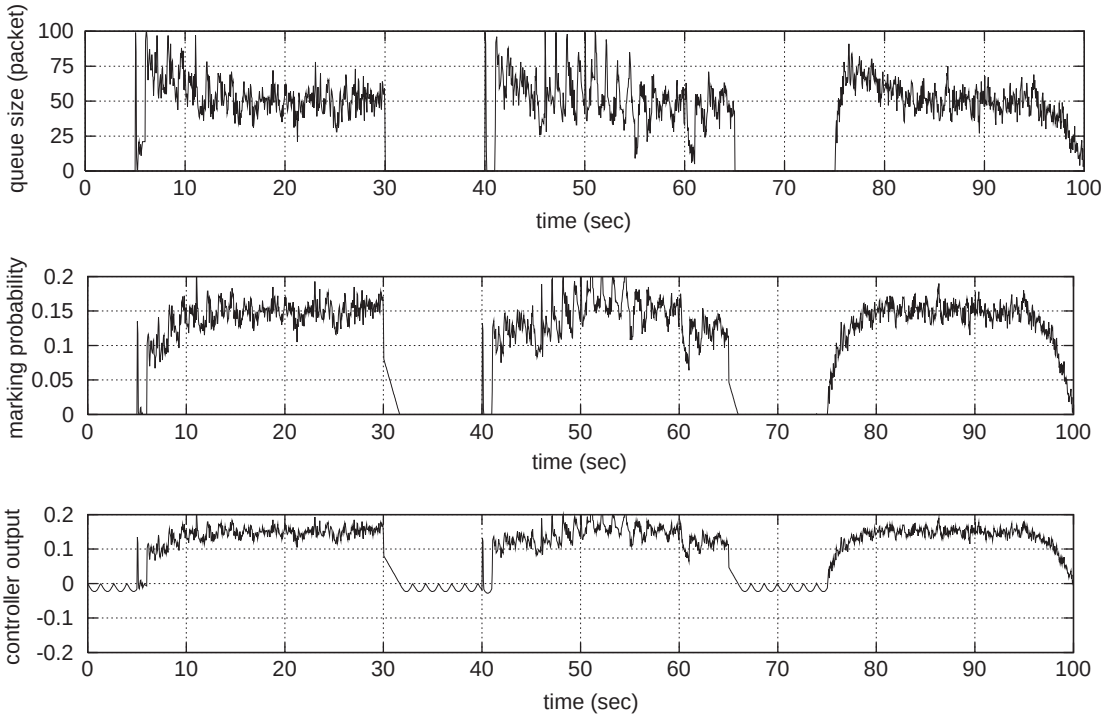

Fig. 7. Experiment 2; PI control with the proposed scheme; $N = 150$, $R = 26$ ms.

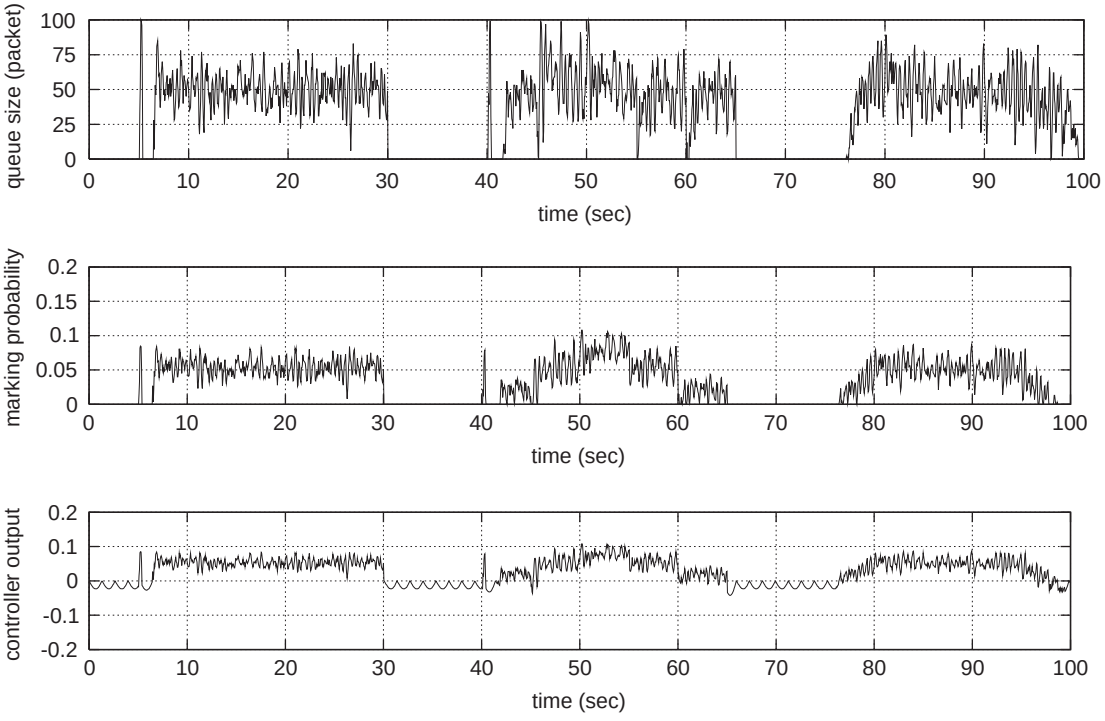

Fig. 8. Experiment 3; PI control with the proposed scheme; $N = 50$, $R = 75$ ms.

show the performance of the proposed scheme with model uncertainty. We use $R = 26$ ms and $N = 150$ in this experiment. The model used for compensator design is no more valid and there exists certain amount of model uncertainties. However, still $R < \bar{R}$ $(= 50)$ and $N > \underline{N}$ $(= 100)$ are satisfied and hence the stability condition is fulfilled. As we can see from Fig. 7, the proposed scheme also works well in this case.

5.3. Experiment 3

In this experiment, we study the performance of the proposed algorithm for the case $R > \bar{R}$ and $N < \underline{N}$. Here we use $R = 75$ ms and $N = 50$. From Fig. 8, we can tell that the proposed scheme still works pretty well. Note that the proposed algorithm in the unsaturated region is exactly same as the conventional PI controller since the states of the dynamic anti-windup compensator will be reset to zero in the unsaturated region. Hence, the performance of the proposed scheme in unsaturated region is influenced by the delay R and will be degraded as R increases.

5.4. Experiment 4

In this experiment, we consider a multiple-node case. As shown in Fig. 9, there are three queues, among which Queue 2 is shared with the main and the cross traffics.

We dynamically establish and terminate the main and cross traffics. All of the cross traffic connections are abruptly on and off at t^c_{on} and t^c_{off} where $t^c_{on} = \{5, 53\}$ and $t^c_{off} = \{47, 100\}$. The number of the cross connections are set to 50. The number of the main traffic connections are randomly increased and decreased during I_{inc} and I_{dec} where $I_{inc} = \{[8, 14],\ [55, 59]\}$ and $I_{dec} = \{[32, 39],\ [94, 100]\}$, respectively. The total number of the main connections is 50.

Figs. 10–12 show the instantaneous queue sizes of Queue 2 for the PI AQM, the incremental algorithm, and the proposed scheme, respectively. Fig. 10 shows a similar result with a single bottleneck case. The windup phenomenon is followed by buffer overflows. Fig. 11 shows that the incremental algorithm works pretty well in this case. However, we can observe the slow response of the incremental algorithm at abrupt traffic change compared to the proposed scheme. From Fig. 12, we can see that the proposed scheme prevents the windup phenomenon and works properly when the connections are established and disconnected abruptly.

Now we compare the performance of the schemes with respect to packet loss ratio, attainable utilization, and average and STD of the queue size of Queue 2. Note that, since we have enabled Explicit Congestion Notification (ECN), the packet loss results entirely from buffer overflows. Fig. 13 shows that packet loss ratio of the PI AQM scheme is much larger than those of the incremental algorithm and the proposed scheme. As already mentioned earlier, this is because the PI AQM scheme does not recover from under-utilization quickly and the queue becomes a drop-tail, which cannot avoid buffer overflows. The attainable utilizations of the incremental algorithm and the proposed scheme are also slightly larger than that of the PI AQM as we can see from Fig. 14. This is because a drop-tail queue usually oscillates severely, which sometimes results in under-utilization. We can also verify the performance of the schemes by comparing the queue behavior. From Fig. 15, we can know that the proposed scheme controls the queue size better than the other algorithms. Also from Fig. 16, we can verify that the queue size variance of the proposed scheme is the smallest among the three algorithms. These implies that the proposed scheme controls the queue size more properly than the conventional PI and the incremental algorithm.

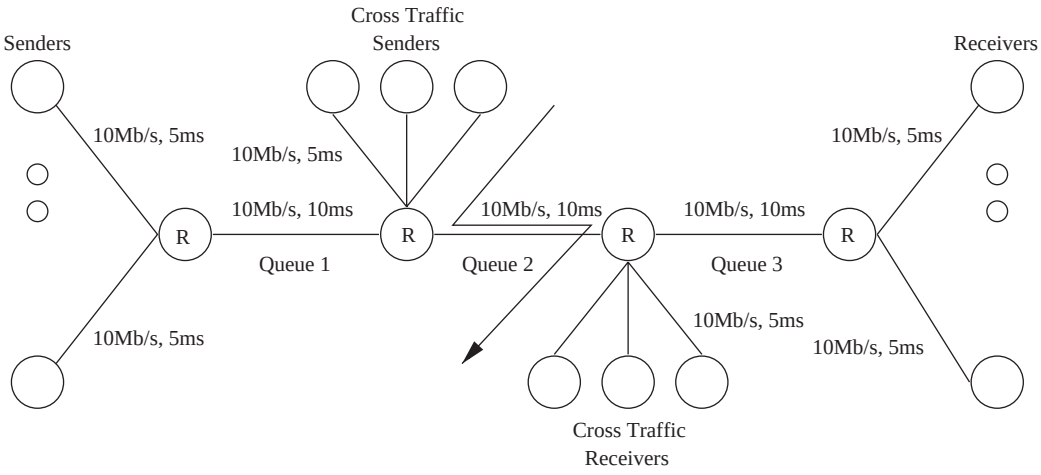

Fig. 9. The multiple-node simulation topology.

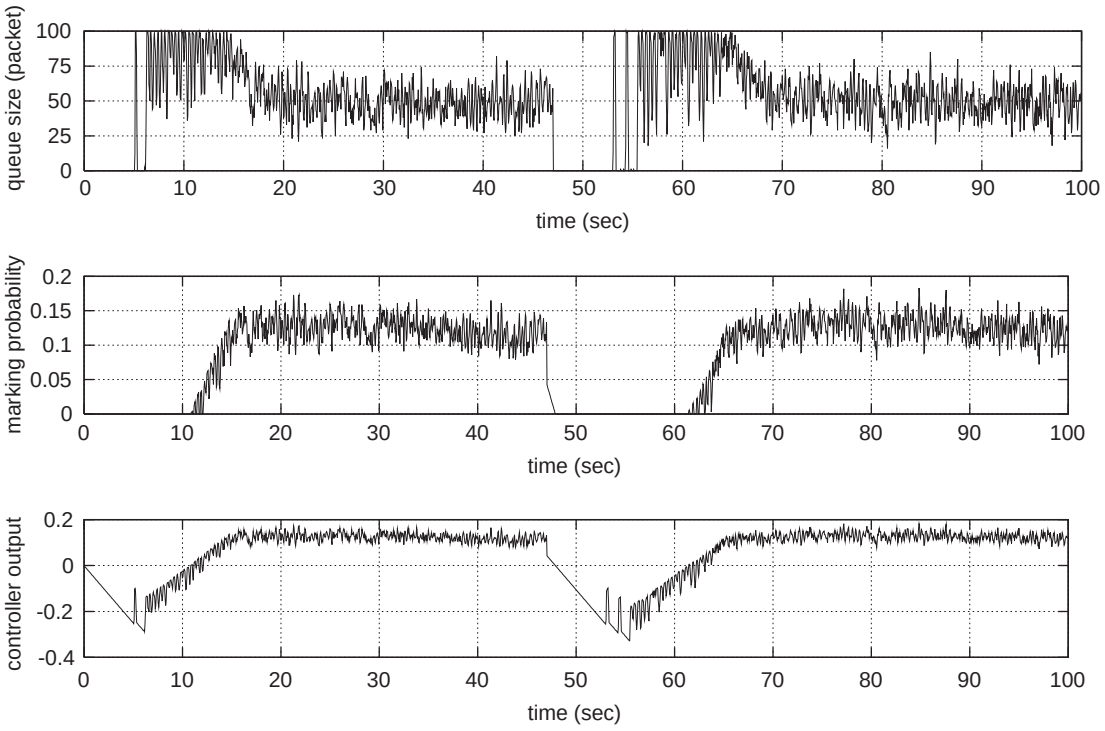

Fig. 10. Experiment 4; Conventional PI control in a multiple-node topology.

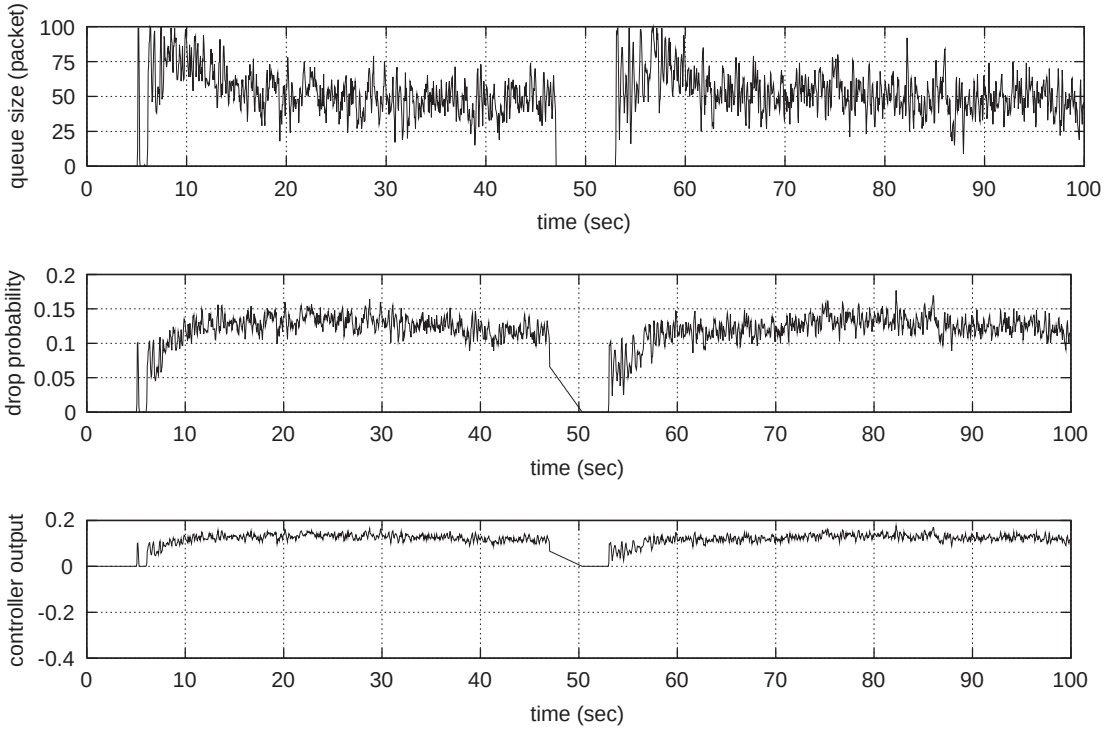

Fig. 11. Experiment 4; PI control with the incremental algorithm in a multiple-node topology.

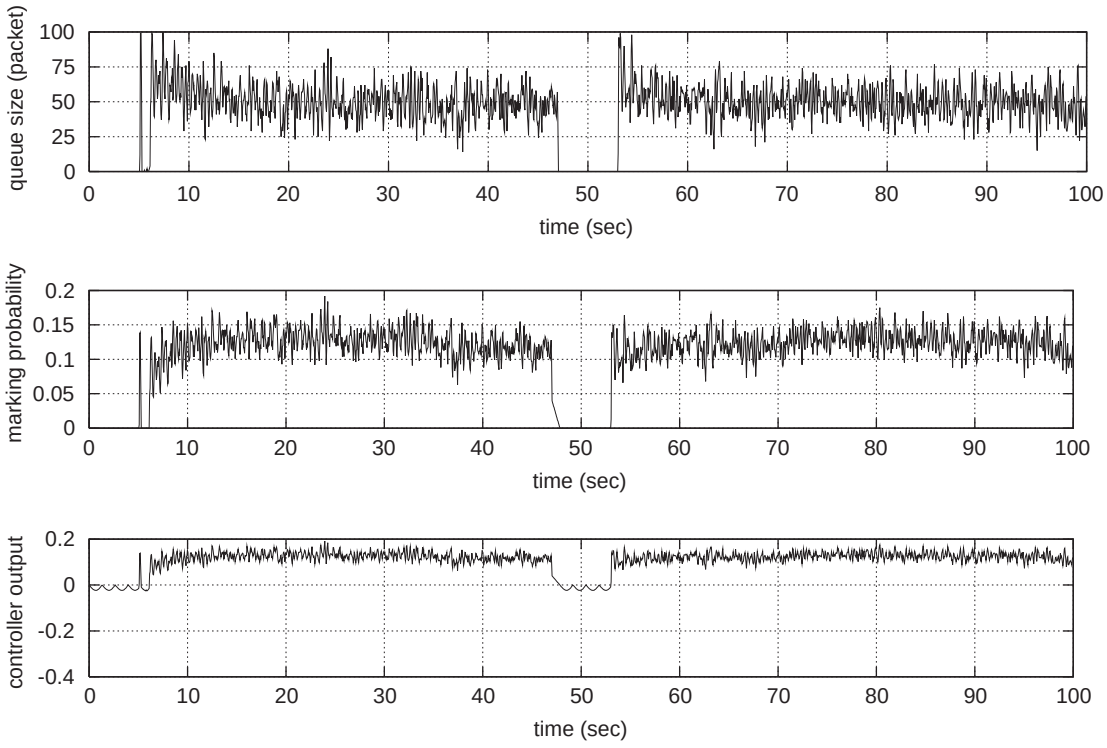

Fig. 12. Experiment 4; PI control with the proposed scheme in a multiple-node topology.

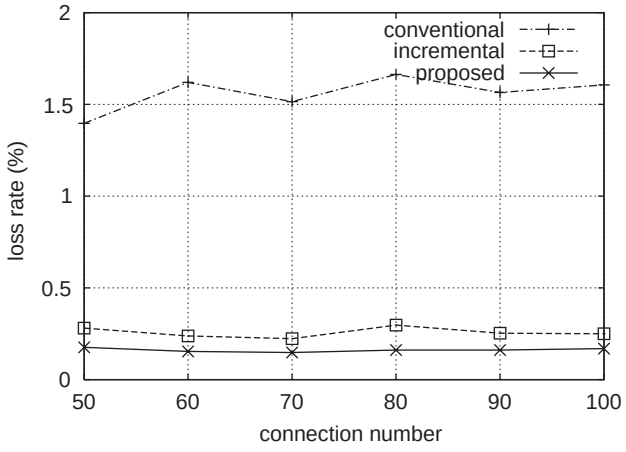

Fig. 13. Experiment 4; Comparison of packet loss rate in a multiple-node topology.

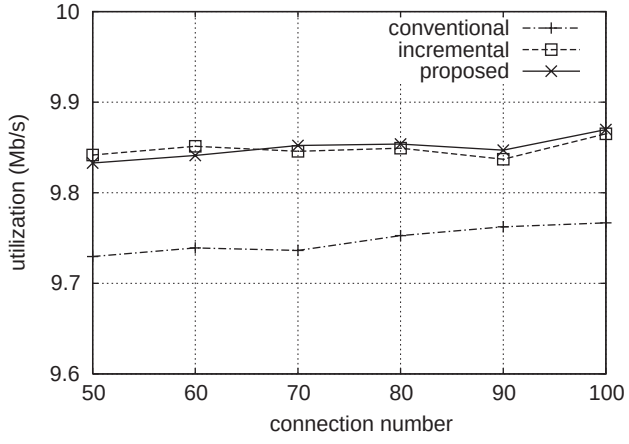

Fig. 14. Experiment 4; Comparison of utilization in a multiple-node topology.

6. Conclusion

The conventional PI AQM scheme outperforms RED significantly. However, the limitation of drop probability can cause significant performance degradation. Because the drop probability is restricted to lie in $[0,1]$, windup phenomenon can happen from this saturation effect. Windup phenomenon leads to degradation of the performance of the conventional PI AQM scheme. To resolve this problem, we applied a dynamic anti-windup scheme based on a TCP/AQM model which includes saturation. We established the total stability and the asymptotic stability without exogenous inputs under model uncertainties for the linearized TCP model in Hollot et al. (2001a). We also compared the proposed scheme with the conventional PI AQM scheme and the incremental algorithm via simulations using an *ns-2*. The simulation results show the effectiveness of the proposed scheme.

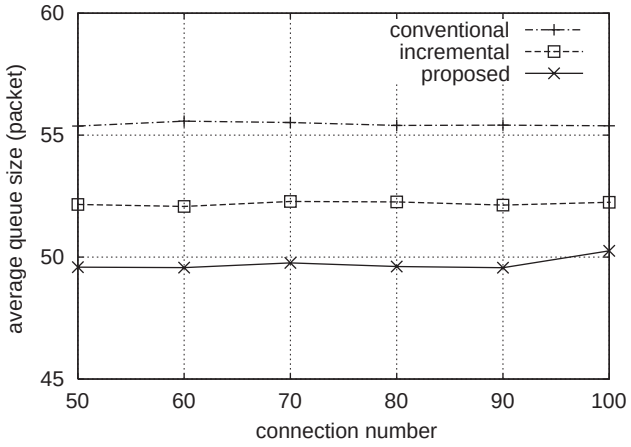

Fig. 15. Experiment 4; Comparison of queue size variation in a multiple-node topology.

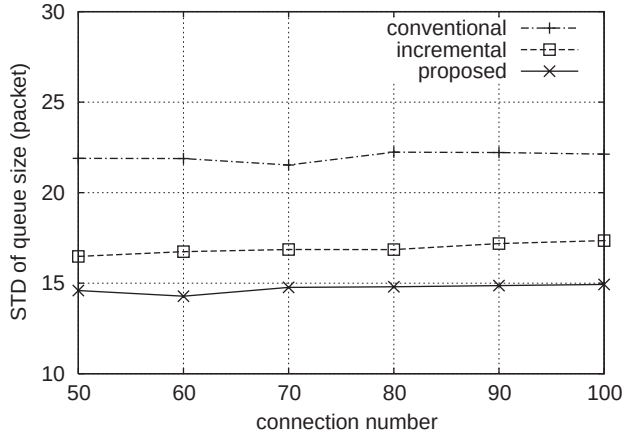

Fig. 16. Experiment 4; Comparison of queue size variation in a multiple-node topology.

Acknowledgements

The authors would like to appreciate Prof. Jong-Koo Park for his helpful comments. This work has been partly supported by the BK21 SNU-UIUC Collaborative Program of the Korean Ministry of Education and Human Resources Development.

References

Astrom, K. J., & Rundqwist, L. (1989). Integrator windup and how to avoid it. In *Proceedings of ACC* (pp. 1693–1698).

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

Aweya, J., Ouellette, M., & Montuno, D. Y. (2001). A control theoretic approach to active queue management. *Computer Networks*, *36*(2–3), 203–235.

Chen, C.-T. (1999). *Linear system theory and design* (3rd ed.). Oxford: Oxford University Press.

Cnodder, S. D., Elloumi, O., & Pauwels, K. (2000). RED behavior with different packet sizes. In *Proceedings of ISCC* (pp. 793–799).

Fall, J., & Varadhan, K. (2001). The ns manual. http://www.isi.edu/nsnam/ns/ns-documentation.

Feng, W., Kandlur, D., Saha, D., & Shin, K. (1999a). *Blue*: *A new class of active queue management algorithms*. Technical Report, UM CSE-TR-387-99.

Feng, W., Kandlur, D., Saha, D., & Shin, K. (1999b). A self configuring RED gateway. In *Proceedings of IEEE INFOCOM*, Vol. 3 (pp. 1320–1328).

Floyd, S., & Jacobson, V. (1992). On traffic phase effects in packet switched gateways. *Internetworking: Research and Experience*, *3*(3), 115–156.

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

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (1995). *Feedback control of dynamic systems* (3rd ed.). Reading, MA: Addison-Wesley.

Franklin, G. F., Powell, J. D., & Workman, M. L. (1990). *Digital control of dynamic systems* (3rd ed.). Reading, MA: Addison-Wesley.

Hollot, C. V., Misra, V., Towsley, D., & Gong, W. (2001a). A control theoretic analysis of RED. In *Proceedings of IEEE INFOCOM*, Vol. 3 (pp. 1510–1519).

Hollot, C. V., Misra, V., Towsley, D., & Gong, W. (2001b). On designing improved controllers for AQM routers supporting TCP flows. In *Proceedings of IEEE INFOCOM*, Vol. 3 (pp. 1726–1734).

Kothare, M. V., Campo, P. J., Morari, M., & Nett, C. N. (1994). A unified framework for the study of anti-windup designs. *Automatica*, *30*, 1869–1883.

Kunniyur, S., & Srikant, R. (2001). Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. In *Proceedings of ACM SIGCOMM* (pp. 123–134).

Lim, H., Park, K.-J., Park, E.-C., & Choi, C.-H. (2002). Proportional-integral active queue management with an anti-windup compensator. In *Proceedings of Conference on Informantion Sciences and Systems*.

Low, S. H., Paganini, F., & Doyle, J. C. (2002). Internet congestion control. *IEEE Control Systems Magazine*, *22*(1).

Mathis, M., Semke, J., & Mahdavi, J. (1997). The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM Computer Communication Review*, *27*(3).

May, M., Bolot, J., Diot, C., & Lyles, B. (1999). Reasons not to deploy RED. In *Proceedings of IWQoS* (pp. 260–262).

Ott, T. J., Lakshman, T. V., & Wong, L. H. (1999). SRED: Stabilized RED. In *Proceedings of IEEE INFOCOM*, Vol. 3 (pp. 1346–1355).

Park, J.-K., & Choi, C.-H. (1995). Dynamic compensation method for multivariable control systems with saturating actuators. *IEEE Transactions on Automatic Control*, *40*, 1635–1640.

Park, J.-K., & Choi, C.-H. (1997). Dynamical anti-reset windup method for discrete-time saturating systems. *Automatica*, *33*, 1055–1072.

Park, J.-K., Choi, C.-H., & Choo, H. (2000). Dynamic anti-windup method for a class of time-delay control systems with input saturation. *International Journal of Robust and Nonlinear Control*, *10*, 457–488.

Peng, Y., Vrancic, D., & Hanus, R. (1996). Anti-windup, bumpless and conditioned transfer techniques for PID controllers. *IEEE Control Systems Magazine*, *16*(4), 48–57.

Stevens, W. R. (1994). *TCP/IP illustrated*, Vol. 1. Reading, MA: Addison-Wesley.

Wang, H., & Shin, K. (1999). Refined design of random early detection gateways. In *Proceedings of GLOBECOM*, Vol. 1b (pp. 769–775).

Yin, Q., & Low, S. (2001). Convergence of REM flow control at a single link. *IEEE Comunication Letters*, *5*(3), 119–121.