

Received September 2, 2020, accepted September 15, 2020, date of publication September 21, 2020, date of current version October 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025680

# Active Queue Management Supporting TCP Flows Using Disturbance Observer and Smith Predictor

RYOSUKE HOTCHI<sup>1</sup>, HOSHO CHIBANA<sup>1</sup>, TAKANORI IWAI<sup>1,2</sup>,  
AND RYOGO KUBO<sup>1</sup>, (Member, IEEE)

<sup>1</sup>Department of Electronics and Electrical Engineering, Keio University, Yokohama 223-8522, Japan

<sup>2</sup>System Platform Research Laboratories, NEC Corporation, Kawasaki 211-8666, Japan

Corresponding author: Ryosuke Hotchi (hotchi.ryosuke@kbl.elec.keio.ac.jp)

This work was supported in part by JSPS KAKENHI under Grant 18K11275.

**ABSTRACT** Active queue management (AQM) is a technique to avoid serious congestion of the transmission control protocol (TCP) flows at a router. AQM based on control theory, which utilizes congestion controllers such as proportional-derivative (PD) controller or proportional-integral-derivative (PID) controller, has been previously proposed. In addition, disturbance observer (DOB) has been utilized to compensate for modeling error of a TCP/AQM congestion control system. However, the DOB-based controllers cannot cope with a large time delay in TCP/AQM networks. Although one of the effective time delay compensators is Smith predictor (SP), the implementation of the DOB and SP in an integrated manner has not been accomplished, because of saturation due to the input limit of packet drop probability. In this paper, a novel TCP/AQM congestion control system with the DOB and SP considering the saturation function is proposed to compensate for the modeling error and time delay simultaneously. Simulation results show that the proposed controller provides better throughput and goodput than conventional controllers. One of the simulations assume that the propagation delay and bottleneck link capacity are set to 100 ms and 100 Mbps, respectively. Under this assumption, it is confirmed that the proposed controller achieves a goodput of 99.55 Mbps whereas the classical PID controller and PD controller with DOB achieve 99.23 Mbps and 99.24 Mbps, respectively.


**INDEX TERMS** Active queue management (AQM), control theory, disturbance observer (DOB), quality of services (QoS), Smith predictor (SP), time delay, transmission control protocol (TCP).

## I. INTRODUCTION

Transmission control protocol (TCP) is often utilized for communications through the Internet, including the communications between sensors and actuators, because of its high reliability. Because the total amount of TCP flows is increasing, their communication packets can cause serious congestion in the routers. When the router is congested, meaning that the buffer of the router is full of packets and it cannot receive any more packets, all the packets arriving thereafter will be dropped until a vacant space is available in the buffer. In communication via loss-based TCPs, the receiver host detects packet loss as a sign of network congestion and notifies the sender host of the congestion. Upon receiving this notification, the sender host scales down the sending window size to control the congestion, resulting

in reduced communication speed. Owing to this mechanism, a large quantity of packets would be dropped if serious network congestion occurs, which leads to the situation where many TCP flows sharing the same network simultaneously reduce their window sizes, thus sharing the network with low throughput. This phenomenon is called global synchronization, and this greatly reduces the communication efficiency, sometimes halting the application services provided over the network [6], [7]. Because of this fact, many studies have been conducted on the efficient method for avoiding congestion at the routers [8].

The congestion control method called active queue management (AQM) has been proposed to deal with this problem [9], [10]. AQM drops packets in the bottleneck router before its buffer becomes full to avoid serious congestion of TCP flows. By dropping the packets before buffer overflow occurs, AQM avoids large queueing delay and global synchronization. One of the most famous AQM methods is

The associate editor coordinating the review of this manuscript and approving it for publication was Hiram Ponce .

random early detection (RED) [11]. RED randomly drops packets with a drop probability determined according to the average queue length. It can keep the queue length relatively stable, preventing global synchronization. Because of its simple calculation procedure, a variety of RED algorithms have been researched and analyzed [12]–[14].

RED has numerous parameters and its parameterization for obtaining satisfactory performance under different circumstances is very difficult [15]. These parameters need to be selected very carefully; otherwise, RED does not perform well, resulting in decreasing throughput and increasing packet loss rate [16]. Therefore, Hollot *et al.* [17] designed the AQM controller on the basis of the control theory. In particular, it is well known that AQM based on the proportional-integral-derivative (PID) control scheme is effective [18], and its controllers were improved by various methods [19]–[22]. In addition, the proportional-integral (PI) controller [23] and proportional-derivative (PD) controller [24] were proposed for TCP/AQM networks to stabilize the queue length around its target value. Compared with a simple algorithm such as RED, AQM based on the control theory tends to have higher throughput [25]. Thus, studies on AQM based on control theory have been actively conducted in recent years [26].

Congestion control based on AQM is negatively impacted by the modeling error of the system and time delay caused by round-trip time (RTT) of the TCP flows. Recently proposed AQM based on control theory is no exception, and there are some studies focusing on compensating either the modeling error or time delay. However, simultaneous compensation of the modeling error and time delay has not been studied for TCP/AQM networks. The related studies on compensation methods for the time delay or modeling error in TCP/AQM networks are described below.

As previously mentioned, the receiver host notifies the corresponding sender host of congestion when it detects the packet loss, and the sender host scales the sending window size down based on it. This means that after sending a packet, the sender host cannot scale the sending window size down until it receives the notification with a specific delay of RTT. Thus, TCP/AQM networks include a time delay element such as RTT between the senders and receivers, which affects the performance of the control system [27]. Several methods have been proposed to compensate for the effect of this built-in delay of the TCP/AQM network, and the most famous method is implementing the Smith predictor (SP) [28], [29]. However, a TCP/AQM congestion control system using an SP cannot cope with fluctuations of the network parameters, such as the number of TCP connections. Because such fluctuations are likely to happen in the TCP/AQM network, this effect must be considered in the design of the AQM scheme.

The robust congestion controller design that uses a disturbance observer (DOB) is an effective way to cope with the parameter fluctuations [30]. DOB has been widely utilized in the field of motion control [31]. It can estimate parameter variations as disturbances, and the estimated disturbance is fed back for robust control. The modeling error can be

included in the disturbance as well, making the modeling much easier. On the other hand, the TCP/AQM congestion control system using the DOB does not have a time delay compensator. The DOB-based controller can be effectively used when the time delay is small. Because of its effectiveness, the DOB has been used for congestion control in AQM with small time delay [32]–[36]. However, a significant time delay in the TCP/AQM network will cause a serious degradation in the performance and result in instability of the network. Therefore, the effects of time delay should be considered in the design of the AQM scheme.

This paper proposes a novel TCP/AQM congestion control scheme supporting TCP flows by implementing disturbance compensation and time delay compensation in an integrated manner, and avoids serious congestions under large RTT. The proposed control system consists of a PD controller, DOB with an artificial delay, and an SP. By utilizing the DOB with an artificial delay, the disturbance of the system can be suppressed even if the TCP/AQM network includes a time delay element. Implementation of the SP will negate the effect of the time delay element by using the plant model. A feature of the SP is that it needs a reasonably approximated plant model to function ideally. Because the disturbance, including the modeling error, can be suppressed by the DOB, the SP can obtain a well-approximated plant model by utilizing the same model employed by the DOB. Kato *et al.* [37] proposed a motion control system with the DOB and SP to compensate for modeling error and time delay. However, unlike the motion control system, the TCP/AQM network congestion control system is subject to input saturation of packet drop probability. The existence of saturation function makes a direct introduction of the DOB and SP impossible. In this study, we take this input saturation into account and designs a novel TCP/AQM network congestion control system with the DOB and SP, to compensate for the disturbance and time delay simultaneously.

The composition of this paper is as follows. The subsequent section describes the TCP/AQM network model. Section III describes the design of the conventional PID controller and PD controller with the DOB. Section IV shows the integrated implementation of the DOB and SP in the classical PD controller by considering the input saturation. Section V shows the effectiveness of the proposed controller based on the simulation results. Finally, our conclusions are presented in Section VI.

## II. TCP/AQM NETWORK MODELS

In this section, a nonlinear TCP/AQM network model [38], a linear model [17], and a nominal model [30] are discussed. The nominal model was utilized for designing controllers in this paper.

### A. NONLINEAR TCP/AQM NETWORK MODEL

AQM is a congestion control mechanism, which attempts to maintain stable queue length by actively dropping packets. The conceptual diagram of a TCP/AQM network is shown

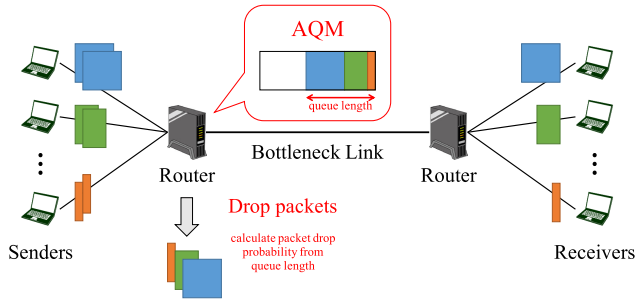


FIGURE 1. TCP/AQM network.

in Fig. 1. As shown in Fig. 1, the bottleneck link router buffers the arriving packets and sends them to the receiver side in first-in first-out (FIFO) manner. The packets that cannot be sent due to bottleneck link capacity limitation remain in the buffer. Thus, the queue length keeps growing if the number of arriving packets is greater than that of the sent packets. AQM drops the arriving packets randomly based on the calculated packet drop probability. By dropping the packets before buffer overflow occurs, AQM avoids global synchronization, which would have happened with DropTail. In addition, the information of packet drop would be notified to the corresponding sender host by the ACK packet sent from the receiver host, resulting in shrinking of the TCP window size and reduction in the number of packets arriving at the bottleneck link router. By these procedures, AQM attempts to maintain the queue length at a constant value.

Misra *et al.* [38] proposed a nonlinear TCP/AQM network model by formulating the TCP window size and queue length dynamics. The nonlinear TCP/AQM network model is shown in (1) and (2),

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t - R(t))}{2R(t - R(t))}p(t - R(t)), \quad (1)$$

$$\dot{q}(t) = -C(t) + \frac{N(t)}{R(t)}W(t). \quad (2)$$

The variables in (1) and (2) are defined as follows:

$W(t) \doteq$  TCP window size [packets],

$q(t) \doteq$  queue length [packets],

$R(t) \doteq$  RTT [s]  $\left( \doteq \frac{q(t)}{C(t)} + T_p \right)$ ,

$C(t) \doteq$  bottleneck link capacity [packet/s],

$T_p \doteq$  propagation delay [s],

$N(t) \doteq$  number of TCP sessions,

$p(t) \doteq$  packet drop probability while  $p(t) \in [0, 1]$ .

### B. LINEAR TCP/AQM NETWORK MODEL

Hollot *et al.* [17] proposed a linear TCP/AQM network model by linearizing the nonlinear model shown in (1) and (2). In

order to linearize (1) and (2), the number of TCP sessions  $N(t)$  and bottleneck link capacity  $C(t)$  were both assumed to be constant, i.e.,  $N(t) \equiv N$  and  $C(t) \equiv C$ . In addition, the operating point where  $\dot{W} = 0$  and  $\dot{q} = 0$  was defined as  $(W_0, p_0, q_0, R_0)$ . From these assumptions, the following equations can be derived:

$$W_0^2 p_0 = 2, \quad (3)$$

$$W_0 = \frac{CR_0}{N}, \quad (4)$$

$$R_0 = \frac{q_0}{C} + T_p. \quad (5)$$

In order to continue the linearization, the dependence of the time delay argument  $t - R(t)$  on the queue length  $q(t)$  is ignored and it is assumed to be fixed at  $t - R_0$ . On the other hand, the dependence of the RTT  $R(t)$  on the queue length  $q(t)$  in the dynamic parameters is retained. As a result, the simplified dynamics are obtained as follows:

$$\dot{W}(t) = \frac{1}{\frac{q(t)}{C} + T_p} - \frac{W(t)}{2} \frac{W(t - R_0)}{\frac{q(t - R_0)}{C} + T_p} p(t - R_0), \quad (6)$$

$$\dot{q}(t) = -C + \frac{N}{R(t)}W(t). \quad (7)$$

By using these assumptions and the simplified dynamics, the TCP/AQM network model for the controller design can be derived as shown in Fig. 2. The specific procedure of this derivation is described in [17]. The transfer function  $P(s)$  can be written as (8)

$$P(s) = -\frac{\frac{C^2}{2N}}{\left(s + \frac{2N}{R_0^2 C}\right) \left(s + \frac{1}{R_0}\right)}. \quad (8)$$

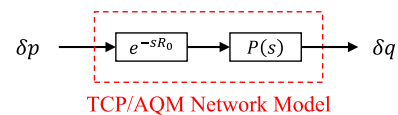


FIGURE 2. Linear TCP/AQM network model for controller design.

### C. NOMINAL TCP/AQM NETWORK MODEL

Based on the linear model shown in Fig. 2, a nominal TCP/AQM network model was proposed in [30]. In this paper, this nominal TCP/AQM network model was utilized to simplify the design of the AQM controller.

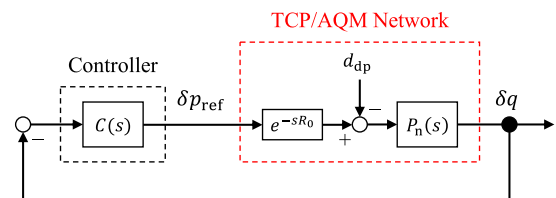


FIGURE 3. Control system with the nominal TCP/AQM network model.

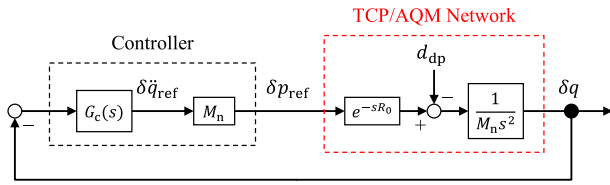


FIGURE 4. Control system with the nominal inertia model.

Figure 3 shows the entire control system with a nominal TCP/AQM network model, where  $P_n(s)$ ,  $C(s)$ ,  $\delta p_{ref}$ , and  $d_{dp}$  denote the nominal TCP/AQM network model, controller, reference packet drop probability, and disturbance in the packet drop probability dimension, respectively. The nominal TCP/AQM network plant model  $P_n(s)$  is defined as (9)

$$P_n(s) = -\frac{C_n^2}{2N_n} \frac{1}{s^2}, \quad (9)$$

where  $N_n$  and  $C_n$  denote the nominal number of TCP sessions and the nominal bottleneck link capacity, respectively. The disturbance  $d_{dp}$  includes the modeling errors due to linearization, nominalization, and coexistence of non-TCP flows such as user datagram protocol (UDP). The notation for input to the TCP/AQM network is changed from  $\delta p$  to  $\delta p_{ref}$  along with the nominalization.

In addition, the inertia model in the TCP/AQM network dynamics  $M_n$  is defined as (10)

$$M_n = -\frac{2N_n}{C_n^2}. \quad (10)$$

From (9) and (10), the following relationship between  $P_n(s)$  and  $M_n$  can be derived:

$$P_n(s) = \frac{1}{M_n s^2}. \quad (11)$$

Using this inertia model  $M_n$ , the block diagram shown in Fig. 3 can be redesigned as shown in Fig. 4, where  $G_c$  denotes a feedback controller such as PD [24] or PID [18], and  $\delta \ddot{q}_{ref}$  denotes the reference queue acceleration. The reference packet drop probability  $\delta p_{ref}$  is calculated as shown in (12).

$$\delta p_{ref} = M_n \delta \ddot{q}_{ref} \quad (12)$$

The feedback controller  $G_c(s)$  calculates  $\delta \ddot{q}_{ref}$  from  $\delta q$ ; the design of the controller is described in the following section.

### III. CONTROLLER DESIGN

In this section, the designs of the PID controller and PD controller with DOB are described. The controllers and DOB are designed based on the nominal TCP/AQM network model shown in Fig. 4. In addition, the operation point of the packet drop probability  $p_0 \approx 0$  is assumed in this study, which implies that  $\delta p = p$ .

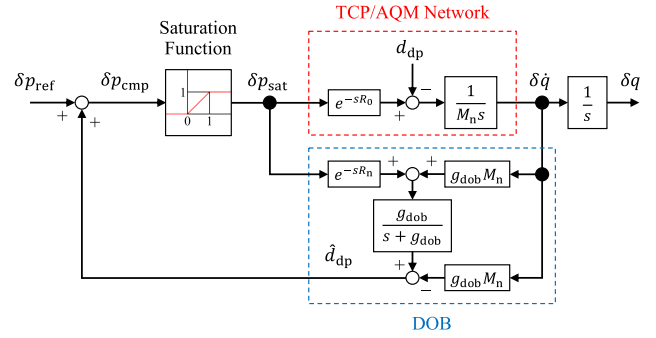


FIGURE 5. TCP/AQM network with the DOB.

#### A. PID CONTROLLER

As shown in Fig. 4, the feedback controller  $G_c(s)$  calculates the reference queue acceleration  $\delta \ddot{q}_{ref}$  from  $\delta q$ . When the PID controller is utilized,  $\delta \ddot{q}_{ref}$  is calculated as (13)

$$\begin{aligned} \delta \ddot{q}_{ref} &= G_c(s) \delta q \\ &= \left( K_p + K_i \frac{1}{s} + K_d s \right) (q_0 - q), \end{aligned} \quad (13)$$

where  $K_p$ ,  $K_d$ , and  $K_i$  denote the proportional gain, derivative gain, and integral gain, respectively. Thus, using (12) and (13), the reference packet drop probability  $\delta p_{ref}$  calculated by the PID controller can be derived as (14).

$$\begin{aligned} \delta p_{ref} &= M_n G_c(s) \delta q \\ &= M_n \left( K_p + K_i \frac{1}{s} + K_d s \right) (q_0 - q). \end{aligned} \quad (14)$$

#### B. PD CONTROLLER WITH DOB

When the PD controller is utilized,  $\delta \ddot{q}_{ref}$  is calculated as (15)

$$\begin{aligned} \delta \ddot{q}_{ref} &= G_c(s) \delta q \\ &= (K_p + K_d s) (q_0 - q). \end{aligned} \quad (15)$$

Thus, using (12) and (15),  $\delta p_{ref}$  calculated by the PD controller can be derived as (16)

$$\begin{aligned} \delta p_{ref} &= M_n G_c(s) \delta q \\ &= M_n (K_p + K_d s) (q_0 - q). \end{aligned} \quad (16)$$

In order to suppress the effect of the disturbance  $d_{dp}$ , we have implemented the DOB [31] as a disturbance compensator. Because the TCP/AQM network has an input delay in its system, the DOB must consider the time delay element [39]. By compensating the disturbance, the DOB attempts to make the actual TCP/AQM network behave identical to the nominal model.

Figure 5 shows the TCP/AQM network with the DOB, where  $\delta \dot{q}$ ,  $\delta p_{cmp}$ , and  $\delta p_{sat}$  denote the queue velocity, the reference packet drop probability including the compensation signal from the DOB, and the reference packet drop probability after the saturation function, respectively. In addition,  $\hat{d}_{dp}$ ,  $R_n$ , and  $g_{dob}$  denote the compensation signal output from the DOB, the nominal time delay, and the cut-off frequency of the DOB, respectively.

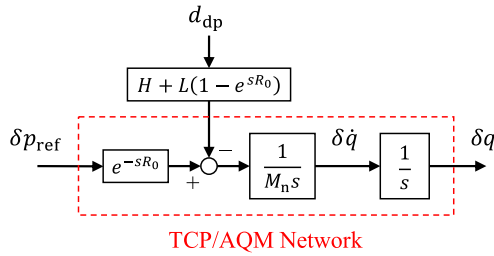


FIGURE 6. Equivalent block diagram of Fig. 5 when  $\delta p_{\text{sat}} = \delta p_{\text{cmp}}$ .

By assuming that  $R_n = R_0$ , the DOB estimates the value of  $d_{dp}$  through the low pass filter, as (17).

$$\hat{d}_{dp} = \frac{g_{dob}}{s + g_{dob}} d_{dp}. \quad (17)$$

The DOB suppresses  $d_{dp}$  by modifying the reference packet drop probability as follows:

$$\delta p_{\text{cmp}} = \delta p_{\text{ref}} + \hat{d}_{dp}. \quad (18)$$

Because the actual input to the TCP/AQM network is the probability, its value must be between 0 and 1. The saturation function will alter the value of  $\delta p_{\text{cmp}}$  based on this rule, as expressed as (19)

$$\delta p_{\text{sat}} = \begin{cases} 0 & (\delta p_{\text{cmp}} < 0) \\ \delta p_{\text{cmp}} & (0 \leq \delta p_{\text{cmp}} \leq 1) \\ 1 & (1 < \delta p_{\text{cmp}}) \end{cases}. \quad (19)$$

By assuming that the error due to saturation does not occur, i.e.,  $\delta p_{\text{sat}} = \delta p_{\text{cmp}}$ , the relationship at the point where the disturbance  $d_{dp}$  joins the control signal can be obtained as (20)

$$\begin{aligned} M_n s \delta q &= e^{-sR_0} (\delta p_{\text{sat}} + \hat{d}_{dp}) - d_{dp} \\ &= e^{-sR_0} \delta p_{\text{cmp}} + e^{-sR_0} \frac{g_{dob}}{s + g_{dob}} d_{dp} - d_{dp} \\ &= e^{-sR_0} \delta p_{\text{cmp}} - \left( \frac{s + g_{dob} (1 - e^{-sR_0})}{s + g_{dob}} \right) d_{dp} \\ &= e^{-sR_0} \delta p_{\text{cmp}} - (H + L (1 - e^{-sR_0})) d_{dp}, \end{aligned} \quad (20)$$

where the high-pass filter  $H = \frac{s}{s + g_{dob}}$  and the low-pass filter  $L = \frac{g_{dob}}{s + g_{dob}}$ . Using (20), the equivalent block diagram of Fig. 5 can be obtained as shown in Fig. 6. Ideally, if  $g_{dob} \rightarrow \infty$  and  $R_0 \rightarrow 0$ , the disturbance is suppressed completely, i.e., the TCP/AQM network model perfectly converges on the nominal inertia model of the TCP/AQM network  $M_n$ . The block diagram of an ideal TCP/AQM network with its disturbance suppressed perfectly is shown in Fig. 7. The robustness of the system with DOB against disturbance has been discussed in previous studies [30], [32].

#### IV. PROPOSED CONTROL SYSTEM

In this section, the implementation of the SP is described and the entire control system, including the PD controller,

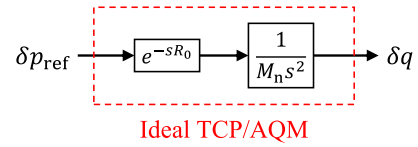


FIGURE 7. Ideal TCP/AQM network with full suppression of disturbance.

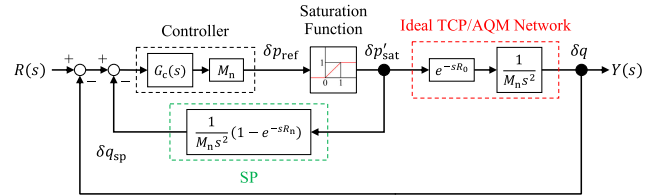


FIGURE 8. Time delay compensation by the SP.

DOB, and SP, is presented. Along with the block diagram of the proposed control system, additional procedure needed for the integrated implementation of the DOB and SP is described. After describing the implementation procedure of the proposed method, the stability analysis using Nyquist diagrams is presented.

#### A. SP

In the feedback control, the control command of the following sampling time is determined according to a deviation of the current command and the response. When a time delay element exists in the feedback loop, the response will be delayed for the same period, inducing a degradation of the control performance. The SP excludes the time delay element outside of the feedback loop by using the control plant model and time delay model [41].

The block diagram of a control system of the ideal TCP/AQM network with SP implemented is shown in Fig. 8, where  $R(s)$  and  $Y(s)$  denote the input and output of the system in the Laplacian domain. In addition,  $\delta p'_{\text{sat}}$  and  $\delta q_{\text{sp}}$  denote the input and output signal of the SP. The signal  $\delta p'_{\text{sat}}$  is also the input signal to the ideal TCP/AQM network. As shown in Fig. 8, the SP utilizes the TCP/AQM model and the delay model for its transfer function. By assuming that the error due to saturation does not occur, i.e.,  $\delta p'_{\text{sat}} = \delta p_{\text{ref}}$ , the transfer function of the entire system can be expressed as follows:

$$\frac{Y(s)}{R(s)} = \frac{G_c(s) e^{-sR_0}}{s^2 + G_c(s) (1 + e^{-sR_0} - e^{-sR_n})}. \quad (21)$$

If the plant model is identical to the actual control plant and the nominal delay  $R_n$  equals the actual delay  $R_0$ , the equivalent block diagram of Fig. 8 can be obtained as shown in Fig. 9. The transfer function of (21) can be also transformed into (22)

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)}{s^2 + G_c(s)} e^{-sR_0}. \quad (22)$$

As Fig. 9 indicates, the time delay element can be excluded from the feedback loop. This means that the characteristic

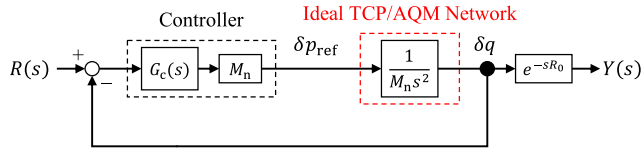


FIGURE 9. Equivalent block diagram of Fig. 8.

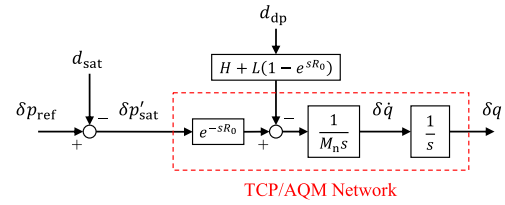


FIGURE 11. Equivalent block diagram of Fig. 10.

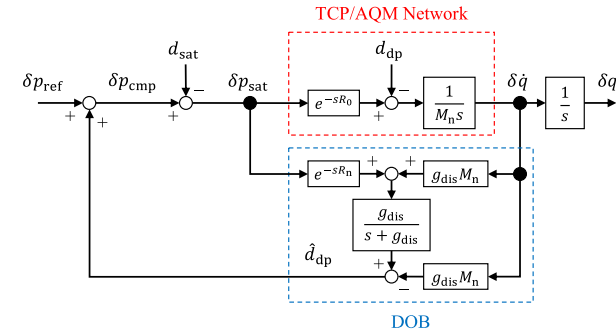


FIGURE 10. TCP/AQM network with the DOB and saturation as a disturbance  $d_{sat}$ .

equation of the control system does not include the time delay element, which will enable the design of a controller without considering the effect of the time delay.

### B. INTEGRATED IMPLEMENTATION OF DOB AND SP

When implementing the DOB and SP in an integrated manner in the TCP/AQM network congestion control system, its nonlinear characteristics prevents the proper functioning of the simple independent implementation of the two compensators. Because the input to the TCP/AQM network is the packet drop probability, the saturation function must occur right before the TCP/AQM network. Because of this design restriction, the compensation signal from the DOB must be added before the input signal passes through the saturation function, as shown in Fig. 5. In the previous section, the saturation function was ignored and it was assumed that  $\delta p_{sat} = \delta p_{cmp}$  whereas, in reality, there is adequate chance that  $\delta p_{cmp}$  becomes lower than 0 or higher than 1 according to (19). The effect of the error occurring at the saturation function can be expressed as (23)

$$\delta p_{sat} = \delta p_{cmp} - d_{sat}, \quad (23)$$

where  $d_{sat}$  denotes the saturation error. By using (23) as an equivalent method of expressing the saturation function, the block diagram shown in Fig. 5 can be redrawn as Fig. 10, and its equivalent block diagram is shown in Fig. 11. The signal alteration caused by the saturation function in Fig. 11 can be expressed as (24)

$$\delta p'_{sat} = \delta p_{ref} - d_{sat}. \quad (24)$$

When the SP is implemented in the TCP/AQM congestion control system with the DOB shown in Fig. 10, the SP requires the saturated input signal without compensation by the DOB,  $\delta p'_{sat}$ , as shown in Fig. 8. However, the required

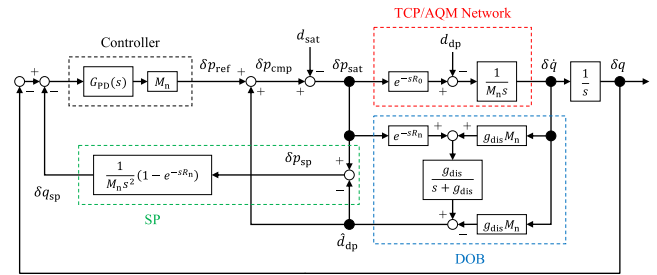


FIGURE 12. Block diagram of the proposed control system.

input  $\delta p'_{sat}$  is not explicitly shown in Fig. 10, since the compensation signal from the DOB  $\hat{d}_{dp}$  is added to  $\delta p_{ref}$  before the saturation occurs. Therefore, our proposed method calculates the required input signal  $\delta p'_{sat}$  based on the actual input signal  $\delta p_{sat}$  and  $\hat{d}_{dp}$  and inputs the estimated  $\delta p'_{sat}$  into the SP.

Using (18), (23), and (24),  $\delta p'_{sat}$  can be rewritten as (25)

$$\begin{aligned} \delta p'_{sat} &= \delta p_{ref} - d_{sat} \\ &= \delta p_{ref} + \hat{d}_{dp} - d_{sat} - \hat{d}_{dp} \\ &= \delta p_{cmp} - d_{sat} - \hat{d}_{dp} \\ &= \delta p_{sat} - \hat{d}_{dp}. \end{aligned} \quad (25)$$

Considering this relationship, the proposed TCP/AQM network congestion control system which consists of the PD controller, DOB, and SP was designed. Figure 12 shows the block diagram of the proposed TCP/AQM network congestion control system, where  $G_{PD}(s)$  denotes the transfer function of PD controller as shown in (15), and  $\delta p_{sp}$  denotes the input signal to the SP, which is expressed as follows:

$$\delta p_{sp} = \delta p_{sat} - \hat{d}_{dp} = \delta p'_{sat}. \quad (26)$$

The input to the SP in the proposed method is equal to the required input signal  $\delta p'_{sat}$ .

### C. STABILITY ANALYSIS USING NYQUIST DIAGRAM

Figures 13 and 14 show the Nyquist diagrams of the loop transfer functions when  $R_n$  values were set to 20 ms and 100 ms, respectively. The proposed PD controller with DOB and SP is denoted as PD + DOB + SP. The conventional PID controller and PD controller with DOB are denoted as PID and PD + DOB, respectively. It was assumed that  $R_0 = R_n$  and  $d_{sat} = 0$ . In addition,  $d_{dp}$  was set to 0 in PID and fully suppressed by DOB in PD + DOB and the proposed PD + DOB + SP. The control gains were set to the same values as those in the simulations. The proportional,

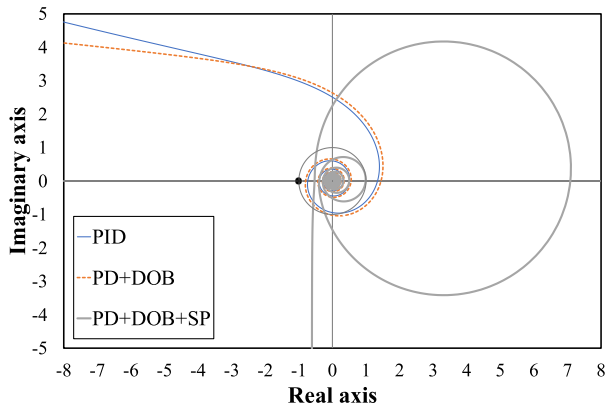


FIGURE 13. Nyquist diagram ( $R_n = 20$  ms).

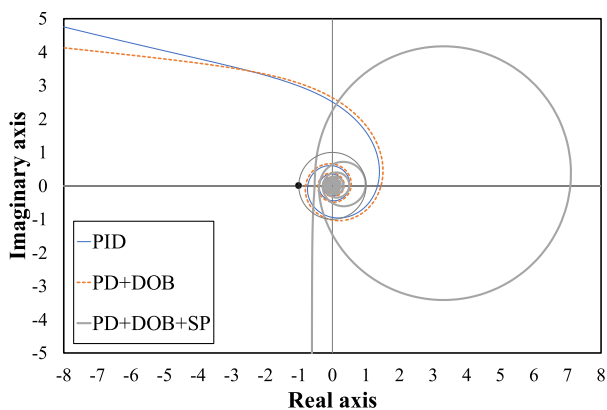


FIGURE 14. Nyquist diagram ( $R_n = 100$  ms).

integral, and derivative gains of PID were set to 900, 700, and 55, respectively. The proportional and derivative gains of PD + DOB and PD + DOB + SP were set to 900 and 60, respectively.

In Fig. 13, all the three methods were stable under this condition of  $R_n = 20$  ms. The proposed PD + DOB + SP method resulted in larger gain and phase margins than any other methods. In Fig. 14, PID and PD + DOB were unstable under this condition of  $R_n = 100$  ms, while the proposed PD + DOB + SP was stable. The results indicated that the proposed PD + DOB + SP effectively compensated for the effect of the time delay element and stabilized the system even under the time delay, which was also evident from the closed-loop transfer function expressed by (22).

### V. SIMULATION

In this section, the performance of the proposed TCP/AQM congestion control system using PD controller with DOB and SP is evaluated based on the simulation results. The results are compared with those of the conventional methods, namely the control systems using RED, controlled delay (CoDel) [42], proportional integral controller enhanced (PIE) [43], PID controller, and PD controller with DOB. For simplicity, the proposed method is denoted as PD + DOB + SP, and

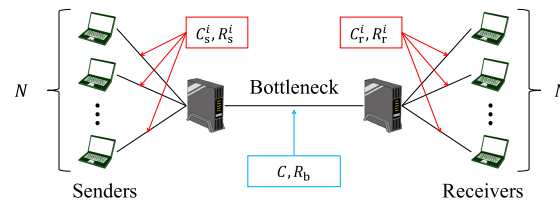


FIGURE 15. Simulation topology.

the five aforementioned conventional methods are denoted as RED, CoDel, PIE, PID, and PD + DOB, respectively.

### A. SIMULATION SETUP

Simulations were performed using the network simulator ns-2. The dumbbell shaped network topology shown in Fig. 15 was utilized in the simulations. In Fig. 15,  $C_s^i$  and  $R_s^i$  denote the link capacity and link latency between the sender side bottleneck link router node and the  $i$ th ( $i = 1, 2, \dots, N$ ) sender node. Similarly,  $C_r^i$  and  $R_r^i$  denote the link capacity and link latency between the receiver side bottleneck link router node and the  $i$ th ( $i = 1, 2, \dots, N$ ) receiver node. The bottleneck link latency is denoted as  $R_b$ ; thus, the propagation delay of the  $i$ th TCP session  $T_p^i$  can be calculated as shown in (27)

$$T_p^i = 2(R_s^i + R_b + R_r^i). \tag{27}$$

Hereon, if the propagation delays of all TCP sessions are equal, the propagation delay is denoted as  $T_p$ .

The network parameters used in the simulations are shown in Table 1, and these are used unless mentioned otherwise. Thus, the propagation delay is set to 100 ms, unless another value is specifically mentioned. In RED [11], the queue weight, minimum threshold, maximum threshold, and maximum packet drop probability were set to 0.002, 50 packets, 150 packets, and 0.02, respectively. In CoDel [44], the interval and target were set to 116 ms and 14.4 ms, respectively. In PIE [45], the update interval, reference latency, and maximum burst allowance were set to 8 ms, 16 ms, and 16 ms, respectively. The parameters for CoDel and PIE were heuristically adjusted so as to have the highest throughput while maintaining the average queue length close to the target queue length of 100 packets. The parameters used in the PD controller, PID controller, and DOB are shown in Table 2. These control parameters were set by referring to [40]. All simulations were performed without using explicit congestion notification (ECN).

### B. QUEUE LENGTH FLUCTUATION

Figure 16 shows the queue length fluctuations of RED, CoDel, PIE, PID, PD + DOB, and PD + DOB + SP for the whole simulation duration of 300 s. The queue length was obtained every 0.05 s. Figure 17 shows the first 25 s of queue length fluctuations, along with the queue length smoothed using exponential moving average (EMA). The smoothing factor for EMA was set to 0.01. The smoothed

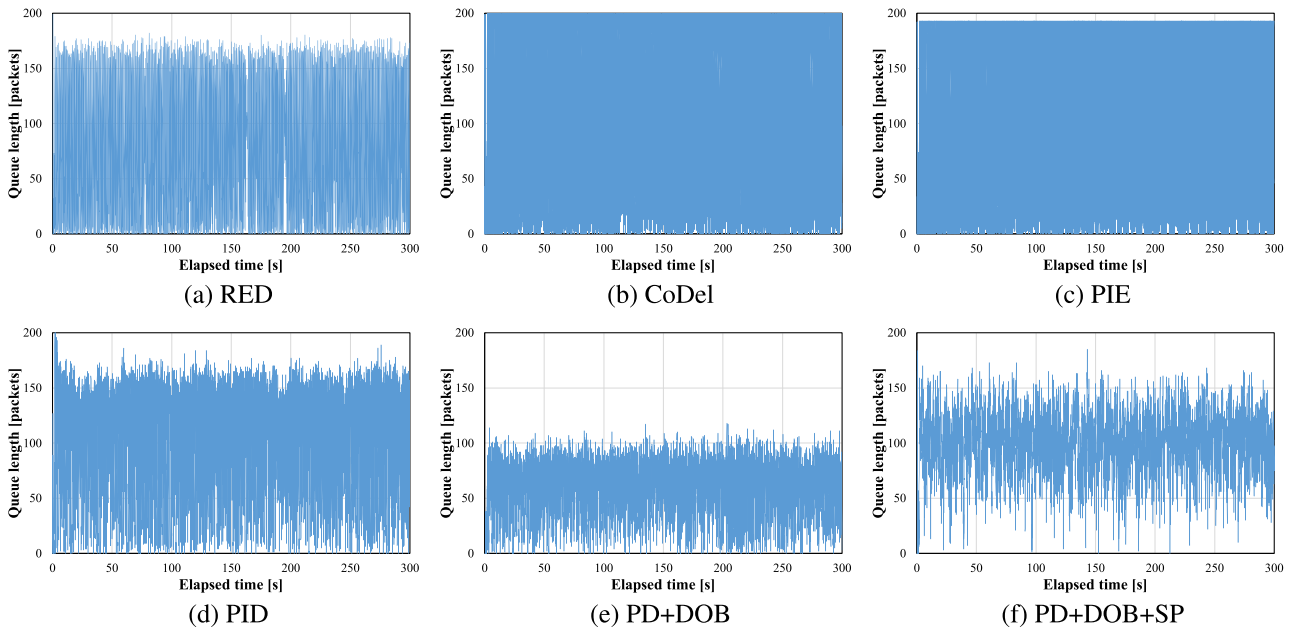


FIGURE 16. Queue length fluctuations.

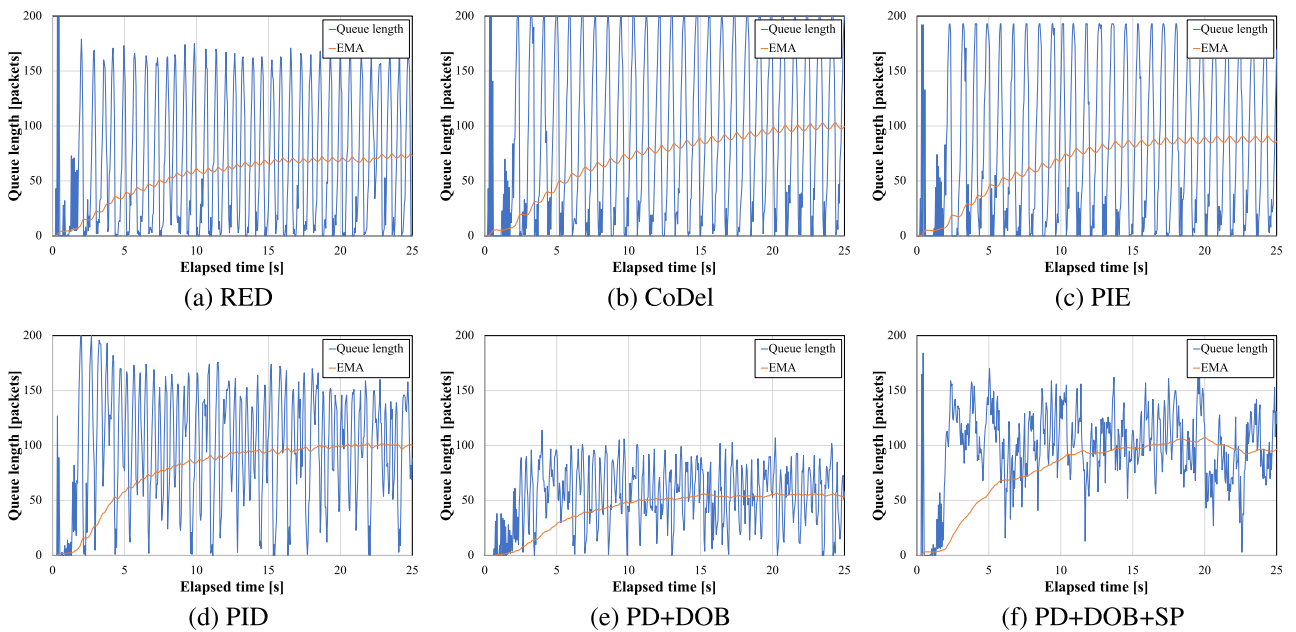


FIGURE 17. Queue length fluctuations during the first 25 s.

queue length was utilized for evaluating the responsiveness of each method.

To evaluate the network performance of each method, the average throughput and goodput were obtained. The average throughput is defined as the time average of total TCP throughput on the bottleneck link. The average goodput is defined as the time average of total throughput excluding retransmission of packets, taking the amount of packet loss into consideration. This excludes the retransmitted packets

from the number of total packets received, denoting how much packet drop occurred.

To evaluate the control performance of each method, the average queue length, standard deviation (SD) of the queue length, maximum queue length, and the total number of empty buffer samples were utilized. The average queue length was measured to evaluate not only the control performance but also average queueing delay. The SD of the queue length was calculated to consider the queue length oscillation and



**TABLE 1. Network parameters.**

Number of TCP sessions $N$	100
$C_s^i$ ( $i = 1, 2, \dots, N$ )	100 Mbps
$C_r^i$ ( $i = 1, 2, \dots, N$ )	100 Mbps
Bottleneck link capacity $C$	100 Mbps
$R_s^i$ ( $i = 1, 2, \dots, N$ )	20 ms
$R_r^i$ ( $i = 1, 2, \dots, N$ )	20 ms
Bottleneck link latency	10 ms
Nominal number of TCP sessions $N_n$	100
Nominal bottleneck link capacity model $C_n$	100 Mbps
Nominal RTT delay model $R_n$	100 ms
Packet size	1040 bytes
Maximum window size $wnd_{max}$	20 packets
Simulation duration time	300 s
Router buffer size	200 packets
Target queue length $q_0$	100 packets
Control period	0.001 s

**TABLE 2. Control parameters for PID controller, PD controller, and DOB.**

$K_p^{pid}$	Proportional gain for PID controller	900
$K_i^{pid}$	Integral gain for PID controller	700
$K_d^{pid}$	Derivative gain for PID controller	55
$K_p^{pd}$	Proportional gain for PD controller	900
$K_d^{pd}$	Derivative gain for PD controller	60
$g_{dif}$	Cut-off frequency for PD and PID controllers	50 rad/s
$g_{dis}$	Cut-off frequency for DOB	50 rad/s

jitter caused by the oscillation. The maximum queue length can be used to consider the occurrence of buffer overflow and maximum queueing delay. The total number of empty buffer samples is defined as the number of measured samples when the queue length is equal to 0, which indicates the duration when the buffer of the bottleneck router is empty. The buffer overflow and empty buffer should be avoided to efficiently utilize the bottleneck link capacity.

To evaluate the responsiveness of each method, the rise time and response delay time of the smoothed queue length were utilized. Both the rise time and response delay time were derived by using EMA of queue length, as shown in Fig. 17. The rise time is defined as the time taken for rising of the smoothed queue length from 10% to 90% of the average queue length. The response delay time is defined as the time taken for rising of the smoothed queue length from the initial value to 50% of the average queue length.

The average throughput, average goodput, average queue length, SD of queue length, maximum queue length, the total number of empty buffer samples, rise time of the smoothed queue length, and response delay time of the smoothed queue length for each method are listed in Table 3. The average throughput, average queue length, SD of queue length, maximum queue length, and the total number of empty buffer samples were measured except for the first 10 s of the simulation duration, to eliminate the surges induced by the TCP algorithm. The ideal behavior of the AQM congestion control

system is to maintain the queue length stable at the target queue length without any fluctuation and fully utilize the bottleneck link capacity.

As listed in Table 3, the proposed PD + DOB + SP showed the highest average throughput and goodput among the six methods, followed by PD + DOB and PID. The relationship between the throughput and goodput, which indicated the amount of packet loss, showed a similar tendency for each method. It was confirmed that the proposed PD + DOB + SP provided the best network performance.

PID had the average queue length closest to 100 packets, while the proposed PD + DOB + SP also kept the average queue length within the range of 100 to 101 packets. This means that PID and PD + DOB + SP provided better control performance with respect to target tracking. However, PID provided a larger SD of the queue length than the proposed PD + DOB + SP, which induced a larger jitter in the queue. In contrast, PD + DOB had the smallest SD of the queue length, followed by the proposed PD + DOB + SP. This means that PD + DOB and PD + DOB + SP provided a smaller jitter. However, PD + DOB provided the worst control performance with respect to target tracking. In addition, only CoDel generated buffer overflow among all the methods. The proposed PD + DOB + SP showed the smallest number of samples with the buffer being empty among all the six methods. It was confirmed that the proposed PD + DOB + SP provided the best control performance with no occurrence of buffer overflow and the least occurrence of empty buffer.

The proposed PD + DOB + SP showed the shortest rise time, followed by PID and PD + DOB. The PID showed the shortest response delay time, followed by the proposed PD + DOB + SP. It was confirmed that PID and the proposed PD + DOB + SP provided better responsiveness than the other four methods.

Therefore, the proposed PD + DOB + SP was proven to be the best from the perspectives of not only network performance but also control performance with respect to target tracking and responsiveness. In the following sections, we mainly present the evaluation of link utilization, which is defined as the ratio of throughput to bottleneck link capacity, since the link utilization is the best parameter representing the network performance of AQM. PID, PD + DOB, and PD + DOB + SP are compared since RED, CoDel, and PIE showed worse results than PID, PD + DOB, and PD + DOB + SP with respect to network and control performances.

### C. CHANGING THE BOTTLENECK LINK CAPACITY

Figure 18 shows the link utilization ratio of the three methods when the bottleneck link capacity  $C$  and the nominal bottleneck link capacity model  $C_n$  were changed simultaneously from 20 to 140 Mbps. The maximum value of  $C$  was set considering the throughput upper bound of the TCP flow. A single TCP session can only send up to 20 packets at once owing to the limitation of the maximum window size shown in Table 1. Because the sender host starts sending the next packet after the ACK is returned from the receiver

TABLE 3. Summary of results in Figs. 16 and 17.

	RED	CoDel	PIE	PID	PD+DOB	PD+DOB+SP
Average throughput [Mbps]	97.11	98.85	98.92	99.63	99.80	99.99
Average goodput [Mbps]	96.76	98.49	98.55	99.23	99.24	99.55
Average queue length [packets]	72.57	100.53	100.62	100.07	57.78	100.28
SD of queue length [packets]	60.71	73.05	70.91	48.45	25.64	27.73
Maximum queue length [packets]	182	200	193	189	118	185
Total number of empty buffer samples	499	306	349	70	51	2
Rise time of the smoothed queue length [s]	11.55	13.65	18.70	8.95	10.15	8.15
Response delay time of the smoothed queue length [s]	5.00	5.20	5.85	4.20	5.05	4.45

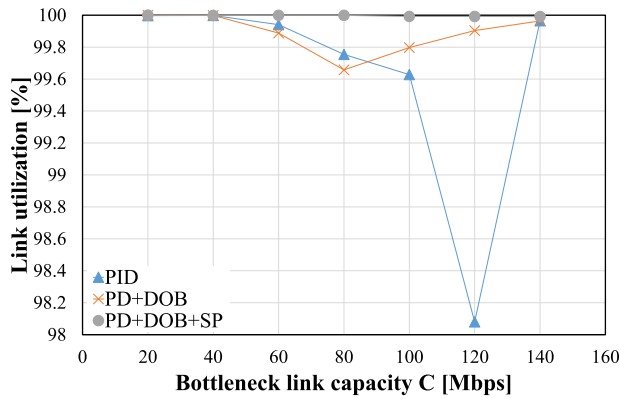


FIGURE 18. Simulation results when C was changed with matching model.

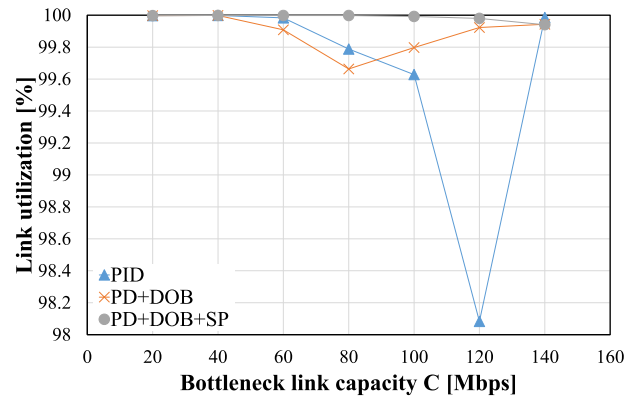


FIGURE 19. Simulation results when C was changed with model mismatch.

hosts, it takes an entire RTT to start sending the second set of 20 packets after the first. In other words, the maximum throughput of a single TCP session in packets/s unit can be derived by dividing this maximum window size by the RTT. The theoretical maximum throughput of the system can be obtained by converting the unit to Mbps and multiplying by the number of TCP sessions. The theoretical maximum throughput of the system can be calculated from the maximum window size  $wnd_{max}$ , packet size, number of TCP sessions  $N$ , and actual RTT  $R$ , as shown in (28)

$$\frac{wnd_{max} * packetsize * 8 * N}{R [s]} \quad [bps]. \quad (28)$$

By using the parameters shown in Table 1 and (28), the possible maximum total throughput can be approximated to 166 Mbps. This value decreases even when the queue exists, due to the queuing delay. Thus, the upper bound value of  $C$  was set to 140 to ensure that the system would not surpass the abovementioned throughput limit.

Figure 19 shows the link utilization ratio when  $C$  was changed while  $C_n$  was maintained at the nominal value of 100 Mbps. This means that the model comprising the controller, DOB, and SP differs in value from the actual network. Hereon, we call this “changing the parameter with model mismatch.”

The link utilization generally drops as  $C$  rises because the inertia of the system increases. This relationship can be confirmed by referring to (10) and (11). However, at a certain

point, this relationship reverses and link utilization starts to increase, as implied by the results of PID and PD + DOB shown in Figs. 18 and 19. This is because  $C$  becomes sufficiently large for the congestion to reduce. On the other hand, the proposed PD + DOB + SP maintains nearly maximum link utilization ratio at any given value of  $C$ , demonstrating its time-delay compensation capability, even with mismatching model value of  $C_n$ .

D. CHANGING THE NUMBER OF TCP SESSIONS

Figure 20 shows the link utilization ratio when the number of TCP sessions  $N$  and the nominal number of TCP sessions  $N_n$  were changed simultaneously from 80 to 200. Figure 21 shows the link utilization ratio when  $N$  was changed while  $N_n$  was maintained at the nominal value of 100. The lower bound of the value of  $N$  was set to 80 owing to its maximum throughput. By using (28), the maximum throughput when  $N = 60$  was calculated to be 99.84 Mbps, which is smaller than  $C$ . Thus,  $N$  was set to values higher than 60 to ensure that the queue would be generated.

The link utilization generally drops as  $N$  drops because of the enlarging inertia of the system. This relationship can be confirmed by referring to (10) and (11). However, at a certain point, this relationship reverses and the link utilization increases, as implied by the results of PID and PD + DOB shown in Figs. 20 and 21. This is because congestion is less likely to occur when  $N$  is sufficiently small. On the

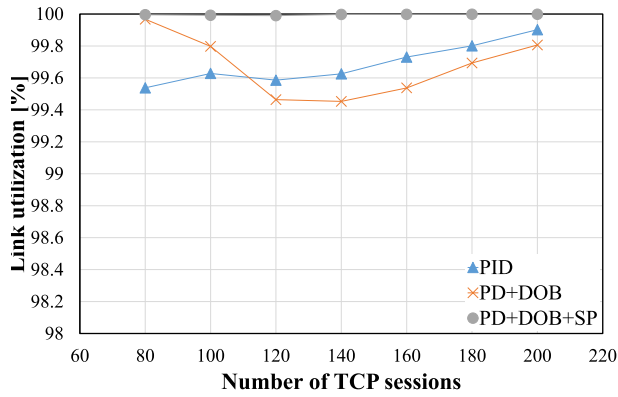


FIGURE 20. Simulation results when  $N$  was changed with matching model.

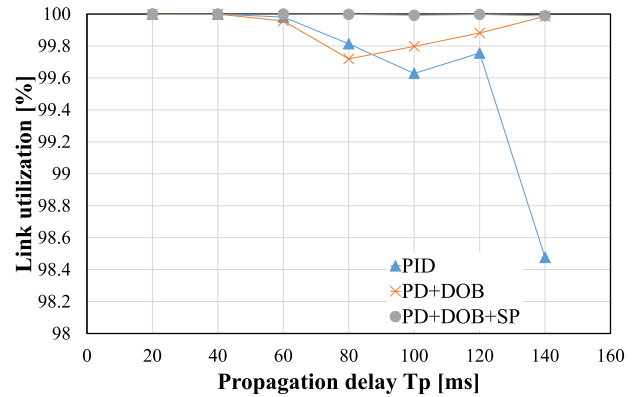


FIGURE 22. Simulation results when  $T_p$  was changed with matching model.

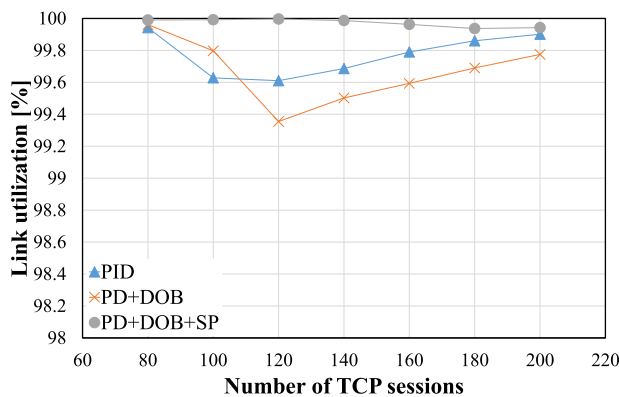


FIGURE 21. Simulation results when  $N$  was changed with model mismatch.

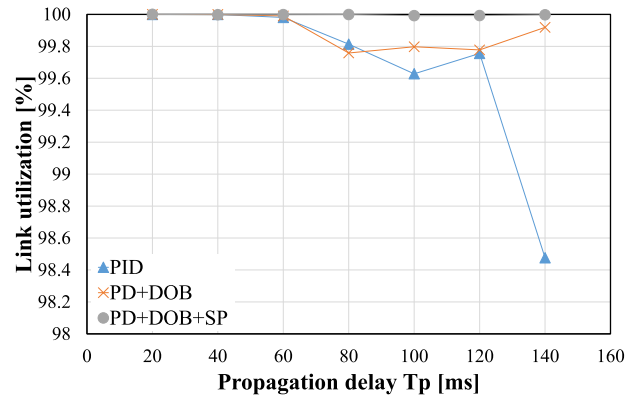


FIGURE 23. Simulation results when  $T_p$  was changed with model mismatch.

other hand, the proposed PD + DOB + SP maintains nearly maximum link utilization ratio at any given value of  $N$ , demonstrating its time-delay compensation capability, even with mismatching model value of  $N_n$ .

**E. CHANGING THE RTT**

Figure 22 shows the link utilization ratio when the propagation delay  $T_p$  and nominal RTT delay model  $R_n$  were changed simultaneously from 20 to 140 ms. The alteration of  $T_p$  was achieved by changing the value of  $R_n^i$  ( $i = 1, 2, \dots, N$ ). Figure 23 shows the link utilization ratio when  $T_p$  was changed while  $R_n$  was maintained at the nominal value of 100 ms. The upper bound of the value of  $T_p$  was set to 140 ms owing to its maximum throughput limit. When  $T_p = 160$ , the maximum throughput was calculated to be approximately 104 Mbps by using (28). At this point, the maximum throughput limit is greater than  $C$  although this is because the effect of queueing delay is ignored. If there are 100 packets of queue when  $C = 100$  Mbps, the queueing delay would be approximately 8 ms. This will lengthen the total RTT, increasing the denominator of (28) and lowering the maximum throughput to 99.04 Mbps. Thus,  $T_p$  was set to values lower than 160 ms to ensure that adequate queue would be generated to discuss the queue control performance.

As Figs. 22 and 23 show, the proposed PD + DOB + SP maintains the highest link utilization ratio among the three methods under any given value of  $T_p$ . PD + DOB tends to generally deliver better performance than PID by compensating the time delay element with the DOB, but its compensation effect appears less effective than that of PD + DOB + SP.

**F. MIXTURE OF UDP FLOWS**

Figure 24 shows the link utilization ratio of all TCP sessions when UDP flows coexist. We assumed that the UDP flows represent the applications and services, such as video and voice, unresponsive to congestion control. The UDP flows can be considered as disturbance for the system. The UDP flows were implemented in the same manner as the TCP sessions shown in Fig. 15, maintaining a simple dumbbell shaped network topology. Each UDP flow utilized in the simulation attempts to send the packets at the constant bitrate of 1 Mbps. Thus, for example, if there are 10 UDP sessions coexisting, the UDP session sources would attempt to send the total load of 10 Mbps to the bottleneck link router. The simulations were conducted with 1, 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95, 98, and 100 UDP sessions.

Unlike TCP, UDP traffic does not have its sending rate limited by the receiving of a packet loss. In fact, the host

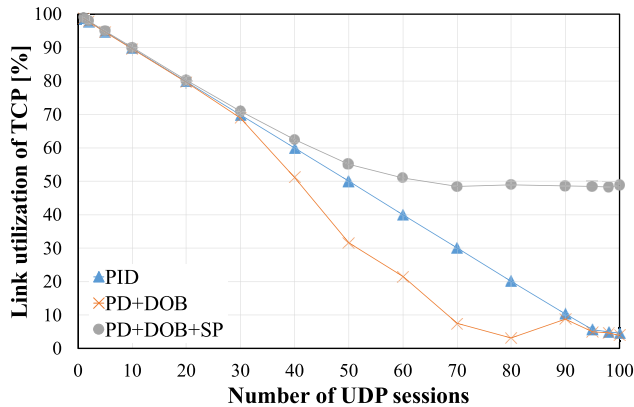


FIGURE 24. Simulation results when UDP flows coexist.

sending the UDP flow does not even wait for the ACK signal to return from the receiver host. Thus, the coexistence of the UDP flows creates a different situation when compared with the network without UDP flows. From Fig. 24, PID and PD + DOB appear to be affected by the different characteristics of the UDP flows, with the total throughput of the TCP flows being extremely low. On the other hand, the proposed PD + DOB + SP succeeded in maintaining a relatively higher throughput than the other two methods. From this result, it can be said that the proposed PD + DOB + SP can maintain higher throughput even when UDP flows coexist.

## VI. CONCLUSION

In this paper, we proposed the AQM congestion control system, which uses a PD controller, DOB, and SP in an integrated manner. The effectiveness of the proposed method was validated by performing simulations using ns-2. The simulations were performed under multiple setups by changing the parameters that mainly affect the behavior of the TCP/AQM network. The simulation results showed that the proposed PD + DOB + SP method generally achieved the highest throughput when compared with the conventional methods. The simulation results with model mismatch also showed that the proposed method maintained its high throughput, demonstrating simultaneous compensation of time delay and model mismatch. In addition, the proposed method maintained a relatively higher throughput of TCP flows than the two conventional methods, PID and PD + DOB, when UDP flows coexisted in the same network. It was confirmed that the proposed PD + DOB + SP has improved performance when compared with the conventional methods. Our future works include the implementation of ECN to improve link utilization and simulations using more complicated scenarios such as those employing multiple bottleneck routers.

Introducing machine learning approaches into AQM is one of the promising solutions to optimize control parameters. If the controller gains, cut-off frequencies, and nominal parameters in the proposed method are adaptively optimized according to network conditions, its performance may be further improved. The real-time optimization of control parameters in the proposed AQM controller is an issue to be addressed.

## REFERENCES

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surv. Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.
- [2] S. Bera, S. Misra, and A. V. Vasilakos, "Software-defined networking for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1994–2008, Dec. 2017.
- [3] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surv. Tuts.*, vol. 17, no. 3, pp. 1294–1312, 3rd Quart., 2015.
- [4] A. Molina-Garcia, J. A. Fuentes, E. Gomez-Lazaro, A. Bonastre, J. C. Campelo, and J. J. Serrano, "Development and assessment of wireless sensor and actuator network for heating and cooling loads," *IEEE Trans. Smart Grid.*, vol. 3, no. 3, pp. 1192–1202, Sep. 2012.
- [5] J. M. Zhang and S. Q. Wang, "Networked control system design and implementation," in *Proc. 1st Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Nov. 2002, pp. 750–753.
- [6] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Comp. Commun. Review*, vol. 18, no. 4, pp. 314–329, Aug. 1988.
- [7] S. Low, F. Paganini, and J. Doyle, "Internet congestion control: An analytical perspective," *IEEE Cont. Sys. Mag.*, vol. 22, no. 1, pp. 28–43, Feb. 2002.
- [8] W. Lautenschlaeger and A. Francini, "Global synchronization protection for bandwidth sharing TCP flows in high-speed links," in *Proc. 16th IEEE Int. Conf. High Perform. Switching Routing (HPSR)*, Jul. 2015, pp. 1–8.
- [9] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, *Recommendations on Queue Management and Congestion Avoidance in the Internet*, document RFC 2309, Apr. 1998.
- [10] R. Adams, "Active queue management: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1425–1476, 3rd Quart., 2013.
- [11] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.
- [12] X. Chen, S. Wong, and C. K. Tse, "Adding randomness to modeling Internet TCP-RED systems with interactive gateways," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 4, pp. 300–304, Apr. 2010.
- [13] S. Patel, B. Gupta, and V. Sharma, "Throughput analysis of AQM schemes under low-rate denial of service attacks," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, Apr. 2016, pp. 551–554.
- [14] S. K. Mohapatra, S. K. Bisoy, and P. K. Dash, "Stability analysis of active queue management techniques," in *Proc. 1st Int. Conf. Man Mach. Interfacing (MAMI)*, Dec. 2015, pp. 1–6.
- [15] W. Chen, Y. Li, and S. Yang, "An average queue weight parameterization in a network supporting TCP flows with RED," in *Proc. IEEE Int. Conf. Netw., Sens., Control (ICNSC)*, Apr. 2007, pp. 590–595.
- [16] S. Woo and K. Kim, "Tight upper bound for stability of TCP/RED systems in AQM routers," *IEEE Commun. Lett.*, vol. 14, no. 7, pp. 682–684, Jul. 2010.
- [17] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. Autom. Control*, vol. 47, no. 6, pp. 945–959, Jun. 2002.
- [18] D. Agrawal and F. Granelli, "Redesigning an active queue management system," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2004, pp. 702–706.
- [19] A. Haider, H. Sirisena, and K. Pawlikowski, "PID based congestion control algorithms for AQM routers supporting TCP/IP flows," *IEICE Trans. Commun.*, vols. E87-B, no. 3, pp. 548–555, Mar. 2004.
- [20] F. Ren and C. Lin, "Speed up the responsiveness of active queue management system," *IEICE Trans. Commun.*, vol. E86-B, no. 2, pp. 630–636, Feb. 2004.
- [21] B. A. Sadek, T. E. Houssaine, and C. Noredine, "A robust PID controller for active queue management framework in congested routers," in *Proc. 1st Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Apr. 2017, pp. 1–6.
- [22] R. Hotchi and R. Kubo, "Remote congestion control using model-free butterfly-shaped perfect delay compensator for active queue management supporting TCP flows," *Nonlinear Theory Appl. (NOLTA), IEICE*, vol. 10, no. 2, pp. 157–172, Apr. 2019.

- [23] Q. Chen and O. W. W. Yang, "On designing self-tuning controllers for AQM routers supporting TCP flows based on pole placement," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 10, pp. 1965–1974, Dec. 2004.
- [24] J. Sun, G. Chen, K. Ko, S. Chan, and M. Zukerman, "PD-controller: A new active queue management scheme," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2003, pp. 3103–3107.
- [25] P. K. Dash and S. K. Bisoy, "Analysis of AQM router of network supporting multiple TCP flows," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, Dec. 2014, pp. 1–5.
- [26] S. K. Bisoy and P. K. Pattnaik, "Design of feedback controller for TCP/AQM networks," *Eng. Sci. Technol. Int. J.*, vol. 20, no. 1, pp. 116–132, Feb. 2017.
- [27] L. Khoshnevisan, F. R. Salmasi, and V. Shah-Mansouri, "Robust queue management for TCP-based large round trip time networks with wireless access link," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2015, pp. 1309–1313.
- [28] H. Ohsaki, H. Yamamoto, and M. Imase, "SPRED: Active queue management mechanism for wide-area networks," in *Proc. Int. Symp. Appl. Internet (SAINT)*, Jan. 2007, pp. 1–7.
- [29] Y. Li, K. Ko, and G. Chen, "A smith predictor-based PI-controller for active queue management," *IEICE Trans. Commun.*, vol. E88-B, no. 11, pp. 4293–4300, Nov. 2005.
- [30] R. Kubo, J. Kani, and Y. Fujimoto, "Advanced Internet congestion control using a disturbance observer," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Nov. 2008, pp. 1–5.
- [31] K. Ohnishi, M. Shibata, and T. Murakami, "Motion control for advanced mechatronics," *IEEE/ASME Trans. Mechatron.*, vol. 1, no. 1, pp. 56–67, Mar. 1996.
- [32] H. Chibana, M. Tadokoro, D. Murayama, K. Suzuki, and R. Kubo, "Robustness evaluation of disturbance observer-based active queue management supporting TCP flows," *IEICE Commun. Express*, vol. 3, no. 10, pp. 311–316, Oct. 2014.
- [33] H. Chibana, M. Yoshino, M. Tadokoro, D. Murayama, K. Suzuki, and R. Kubo, "Disturbance-observer-based active queue management with time delay using software-defined networking controller," in *Proc. 41st Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2015, pp. 1049–1054.
- [34] M. Lin, T. Ren, H. Yuan, and M. Li, "The congestion control for TCP network based on input/output saturation," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 1166–1171.
- [35] P. Wang, C. Zhu, and X. Yang, "A novel AQM algorithm based on feed-forward model predictive control," *Int. J. Commun. Syst.*, vol. 31, no. 12, article e3711, Aug. 2018.
- [36] L. Khoshnevisan, X. Liu, and F. R. Salmasi, "Predictive sliding-mode congestion control for wireless access networks with singular and non-singular control gain," *IET Control Theory Appl.*, vol. 14, no. 13, pp. 1722–1732, Aug. 2020.
- [37] A. Kato, A. Muis, and K. Ohnishi, "Robust network motion control system based on disturbance observer," *Automaika*, vol. 47, nos. 1–2, pp. 5–10, Jan. 2006.
- [38] V. Misra, W. Gong, and D. Towsley, "A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 4, pp. 151–160, Oct. 2000.
- [39] K. Hong and K. Nam, "A load torque compensation scheme under the speed measurement delay," *IEEE Trans. Ind. Electron.*, vol. 45, no. 2, pp. 283–290, Apr. 1998.
- [40] R. Kubo, J. Kani, and Y. Fujimoto, "Congestion control in TCP/AQM networks using a disturbance observer," (in Japanese), *IEEJ Trans. Ind. Appl.*, vol. 129-D, no. 6, pp. 541–547, Jun. 2009.
- [41] O. J. M. Smith, "A controller to overcome dead time," *ISA J.*, vol. 6, no. 2, pp. 28–33, Feb. 1959.
- [42] K. Nichols and V. Jacobson, "Controlling queue delay," *ACM Queue*, vol. 10, no. 5, pp. 1–15, May 2012.
- [43] R. Pan, P. Natarajan, C. Pignone, M. S. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg, "PIE: A lightweight control scheme to address the bufferbloat problem," in *Proc. 14th IEEE Int. Conf. High Perform. Switching Routing (HPSR)*, Jul. 2013, pp. 148–155.
- [44] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, *Controlled Delay Active Queue Management*, document RFC 8289, Jan. 2018.
- [45] R. Pan, P. Natarajan, F. Baker, and G. White, *Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem*, document RFC 8033, Feb. 2017.



**RYOSUKE HOTCHI** received the B.E. degree in electronics and electrical engineering and the M.E. degree in integrated design engineering from Keio University, Japan, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in integrated design engineering.

His research interests include network control and queue control systems. He is a Student Member of the Institute of Electronics, Information and Communication Engineers (IEICE).



**HOSHO CHIBANA** received the B.E. degree in electronics and electrical engineering and the M.E. degree in integrated design engineering from Keio University, Japan, in 2014 and 2016, respectively. In 2016, he joined Kansai Electric Power Co., Inc., Osaka, Japan.



**TAKANORI IWAI** received the B.E. and M.E. degrees in electrical and electronic engineering from Shinshu University, Japan, in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree in integrated design engineering with Keio University. In 2004, he joined NEC Corporation, Japan, and is currently a Principal Researcher at System Platform Research Laboratories. His research interests include system control, mobile and wireless networking, and IoT service networks.

He received the Electrical Science and Engineering Award from the Promotion Foundation for Electrical Science and Engineering in 2016, and the Best Paper Award from the IEEE ComSoc International Communications Quality and Reliability Workshop (CQR '17).



**RYOGO KUBO** (Member, IEEE) received the B.E. degree in system design engineering and the M.E. and Ph.D. degrees in integrated design engineering from Keio University, Japan, in 2005, 2007, and 2009, respectively.

In 2007, he joined the NTT Access Network Service Systems Laboratories, NTT Corporation, Japan. Since 2010, he has been with Keio University, Japan, where he is currently an Associate Professor with the Department of Electronics and Electrical Engineering. From 2019 to 2020, he also held the position of Honorary Research Fellow at the Department of Electronic and Electrical Engineering, University College London (UCL), U.K. His research interests include system control, optical communications, networking, and cyber-physical systems. He is a member of the Optical Society (OSA), the Institute of Electrical Engineers of Japan (IEEJ), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Society of Instrument and Control Engineers (SICE). He received the Best Paper Award from the IEICE Communications Society in 2011, the IEEE International Conference on Communications (ICC '12) Best Paper Award in 2012, the Leonard G. Abraham Prize from the IEEE Communications Society in 2013, and the 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR '18) Best Paper Award in 2018.

...