IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 24, 2020, accepted January 22, 2021, date of publication February 3, 2021, date of current version February 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056885

A Self-Tuning Congestion Tracking Control for TCP/AQM Network for Single and Multiple Bottleneck Topology

SUKANT KISHORO BISOY[1], PRASANT KUMAR PATTNAIK[2], (Senior Member, IEEE), MANGAL SAIN[3], AND DO-UN JEONG[3]

[1]Department of Computer Science and Engineering, C.V. Raman Global University, Bhubaneswar 752054, India
[2]School of Computer Engineering, KIIT University, Bhubaneswar 751024, India
[3]Division of Computer Engineering, Dongseo University, Busan 47011, South Korea

Corresponding authors: Mangal Sain (mangalsain1@gmail.com) and Do-Un Jeong (dujeong@dongseo.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2018R1D1A1B07045337.

ABSTRACT In this work a self-tuning rate and queue based proportional and integral controller called SRQ-PI is proposed to efficiently control the queue length with small overshoot and faster settling time. SRQ-PI proposes a new control tracking function that maps level of congestion to the packet drop probability dynamically. In SRQ-PI, the incoming traffic rate is estimated and used with the proportional and integral controller. The SRQ-PI tunes itself and stabilizes the system with internal feedback without requiring any external feedback. Furthermore, the stability of the SRQ-PI is analyzed using control theory and presents systematic guidelines to select the control gain parameters. NS2 is used to carry out the simulation work. The simulation result demonstrates that SRQ-PI is stable and gets faster transient response due to lower average delay jitter and robust against dynamic network parameters. The SRQ-PI outperforms proportional integral (PI), Intelligent adaptive PI (IAPI) and Random exponential marking (REM) algorithm.

INDEX TERMS Feedback control, self-tuning, control theory, stability, transient response.

I. INTRODUCTION

Participation of Million people over Internet created network congestion problem. Congestion brings long delay of the data packets in sending and receiving and wastes resources due to data packet dropping. Traditional method of buffer management at the Internet router drops the packets when the buffer is full. This shows that passive nature of management increases the round trip time (RTT) of each packet and creates TCP synchronization problem [1]. On the other hand, use of active queue management (AQM) [2] technique at bottleneck router can efficiently control the traffic passing through the router and avoids TCP synchronization problems and achieves better bandwidth utilization. Normally in smart grid application, managing heterogeneous flow suffers from communication delay. Each industrial router is configured with buffering with different capacity which needs to handle incoming arriving packets. On the other hand it is important to maintain good link utilization with lower delay. To control the queue length of router buffer, we require a stable AQM technique which can handle all the incoming traffic with small overshoot and small delay.

Moreover, Internet Engineering Task Force (IETF) recommended use of AQM overcome the performance limitation of TCP. The objective of AQM is to signal congestion early by dropping the packets or marking the packets before the buffer is full.

Broadly AQM techniques can be categorized into different groups based on network (wired or wireless), their control function or network (wired or wireless) or congestion measure used [3]. Considering the congestion measure, the AQM techniques can be classified into three groups. The first group of AQM is based on rate of incoming traffic. Adaptive virtual queue (AVQ) [4], BLUE [5] and GREEN [6] can be treated as rate-based AQM as it uses incoming traffic rate for congestion measure. The second group is based on the current queue length. For example, Random early detection (RED) [7], Proportional Integral (PI) [8], Proportional Derivative (PD) [9] and Proportional Integral Derivative (PID) [10] can be considered as queue based techniques as queue length is

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

used for the congestion measure. In the third group, both incoming traffic rate and queue length are used to measure congestion and determine packet dropping probability. Random Exponential Marking (REM) [11], RaQ [12], Virtual Rate Control (VRC) [13] and Yellow [14] belongs to third group. RED is the most popular and widely used technique recommended by IETE. Based on RED other mechanisms such as Adaptive RED (ARED) [15], PD-RED [16], Refined (Re-ARED) [17], Nonlinear RED (NL-RED) [18], Stabilized RED (SRED) [19], Cautious RED (CARED) [20], Loss-ratio based RED (LRED) [21], Exponential RED [22] and Dynamic RED (DRED) [23] have been proposed. Although RED and PD-RED can overcome the global synchronization and blocking problem, it can have difference in performances based on the setting of parameters. The difficulty lies in parameter setting for stabilizing the system under dynamic network scenarios. To overcome such difficulties, several mechanisms such as Adaptive RED, PD-RED, and PI have been proposed. The heuristic approaches such as RED and PD-RED achieves better performance by proper adjustment of its parameter. There is no systematic method available to select control gain parameter to bring stability in the network with lower average jitter. Hence, guidelines to select control gain with theoretical analysis is required in dynamic scenarios where parameters such as the number of TCP sessions, bandwidth, and round trip time (RTT) traffic keep changing. It should work well in single network and multi-bottleneck topology. Recently many active queue management techniques have been proposed [24]. As the Internet traffic keeps changing designing an efficient AQM technique for this dynamic scenario and achieving stable and faster responsive AQM controller is a challenging task.

In this paper, we propose a self-tuning rate and queue based PI controller named SRQ-PI to control the queue length at the AQM router supporting TCP flows.

The contributions of this paper are as follows:

- In this paper, we propose a self-tuning rate and queue based proportional and integral (SRQ-PI) technique as a feedback process to control the queue length efficiently at the AQM router supporting TCP flows.
- Stability of SRQ-PI is analyzed using two different methods:

  (1) Using classical control theory and
  (2) Simulation method.
- It provides systematic theoretical guidelines for selection of control gain parameters for SRQ-PI.
- The proposed controller is stable and achieves faster transient response and robust under different network parameters.
- Finally, proposed work is simulated using NS2 and compared with existing schemes under various scenarios.

Proposed work is compared with many existing techniques developed from 2018 to 2020 and summarized in Table 1.

Remaining sections are organized as follows. Section 2 summarizes the related work and section 3 presents dynamic models. Section 4 explains proposed SRQ-PI algorithm where stability is analyzed and selection of parameter, simulation setup and results are analyzed. Finally, the conclusion of the work is given section 5.

TABLE 1. Comparison of existing techniques.

AQM Technique	Year	Plant Model	Stability	Settling time	Link Utilization	Average Delay
PD+DOB+SP [25]	2020	Linearized TCP/AQM model	Average	Very high	Very Good	Average
RED_E [26]	2020	-	-	High	Average	High
OBC [27]	2019	Linearized TCP/AQM model	Average	High	-	-
PDC [28]	2019	Linearized TCP/AQM model	Good	Less	Average	High
LFC [29]	2018	Linearized TCP/AQM model	Good	Very Low	-	-
CoDel [30]	2018	Linearized TCP/AQM model	Average	High	Good	High
FF-AQM [31]	2018	Linearized TCP/AQM model	Good	Average	-	-
SRQ-PI (Proposed technique)		***Linearized TCP/AQM model***	***Very Good***	***Very Low***	***Good***	***Low***

II. RELATED WORK

Droptail creates Lock-out, and global synchronization [1] problems due to its passive nature. To avoid such problem, AQM can be used as a router based technique. RED is one of the widely used AQM techniques recommended by IETE. Dropping probability (p) of RED is defined as given in Eq. (1).

$$p = p_{max} \times \frac{avg - min_{th}}{max_{th} - min_{th}} \tag{1}$$

where p_{max} is the maximum value of the p. The max_{th} and min_{th} are the maximum and minimum threshold value respectively. It calculates average queue length denoted by avg and compare with max_{th} and min_{th} to decide the packet drop probability. All the incoming packets are dropped with probability 1, when avg value exceeds max_{th}. When the value of avg lies between min_{th} and max_{th}, the packet is being dropped/marked with the probability given in Eqs.(1). However, no packet is being dropped when avg is less than min_{th}. The difficulty lies in the parameter setting for stabilizing the system where the network environment is dynamic. To address such problems, several mechanisms such as self-configuring RED, Adaptive RED, PD-RED, and PI are proposed.

The performance of Adaptive RED is better than self-configuring RED and dropping/marking probability function of ARED is defined as follows:

$$p = \begin{cases} p+\alpha, & \text{if } avg \succ Q_{ref} \text{ and } p \leq 0.5; \\ p\times\beta, & \text{if } avg < Q_{ref} \text{ and } p \geqq 0.01; \end{cases} \quad (2)$$

where α is the increasing factors and β is the deceasing factors used in packet drop probability function depending upon the condition mentioned in Eq.(2). PD-RED technique is proposed based on the two factors: one is queue length error and second one is change in error to adopt p_{max} value. It uses control theory for system stability. The dropping probability function is defined as in Eq. (3).

$$p(t) = p(t-1) + K_p \frac{\left(q_{avg}(t) - q_{ref}\right)}{B} + K_d \frac{\left(q_{avg}(t) - q_{avg}(t-1)\right)}{B}; \quad (3)$$

where, K_p is the proportional gain and K_d is the derivative gain of PD-RED controller. q_{ref} is reference in queue length and B is the buffer size. To adopt dropping probability of PD-RED, it considers only the current information such as error in average queue length and differential error. However, the past history is not considered due to absence of integral part which makes the system unstable under the presence of UDP traffic, short-lived and HTTP traffic. However, the performance of PD-RED is better than Adaptive RED.

PI is proposed as a queue based AQM controller based on classical control theory whose performance is better than RED. For each sampling period (t), the packet drop probability is updated as in Eq.(4).

$$p(t) = p(t-1) + a\left(q(t) - Q_{ref}\right) - b\left(q(t-1) - Q_{ref}\right); \quad (4)$$

where a and b are the proportional and integral constants. PI controller enhances the responsiveness of TCP/AQM system [32]. However, the settling time is very high and the responsiveness is very slow in PI controller. REM measure congestion based on both queue length and incoming traffic. Based on the queuing information and incoming rate r (t), the price function μ (t) is defined as in Eq. (5).

$$\mu(t) = \mu(t-1) + \gamma\left(\alpha\left(q(t) - Q_{ref}\right) + r(t-1) - C\right); \quad (5)$$

Then price function is used to find packet marking probability function p (t) as defined in Eq.(6).

$$p(t) = 1 - \phi^{-\mu(t)}; \quad (6)$$

III. THE DYNAMIC MODEL

Many theoretical models were developed in the literature for the network flow dynamics using TCP traffic. The working behavior of TCP modeled and analyzed using differential equation [2]. It assumes all the homogeneous TCP flows connected to a single bottleneck topology having the same delay as shown in Fig. 1.

Also, it is assumed that the window size of the source remains unchanged for all the times. All the source nodes communicate their corresponding destination through the intermediate router and the data will be buffered in the router while passing through it. Based on current queue length the congestion is estimated by the router by computing the packet drop probability (p) and provides feedback to the end point. However, the router provides feedback through the acknowledgement generated by the receiver. Then the sender window size is adjusted at next time slot based on the feedback of the router. Using control theory notion a network can be modeled as "feedback control system". Here the endpoint (TCP Process) and router can form a "plant" controlled by the controller. In this, AQM techniques can be deployed as controller to control the plant. To control the source rate of endpoints it uses packet dropping probability function as a control signal to the plant. Fig. 2 shows the working principles of AQM technique using well know control theory technique. As depicted in Fig.2 the queue length is considered as controlled variable and Q_{ref} is considered as target queue length (reference input). Then error signal is estimated based on the current queue length and target and used to control the plant. In last two decades, many models have been presented to understand the dynamics of TCP/AQM system. The fluid flow model MGT is initially presented in [8] and reviewed later without considering any AQM controller where a non-negative matrix method is proposed to model the dynamics of TCP system [33]. The analytical model for TCP/AQM system is presented in [2]. Later the Model Predict Control (MPC) [34, 35] is designed to predict the dynamics of the system and to optimize the control signal during each sampling period. And further the accuracy of the MGT fluid flow model is improved in [36] under massive TCP traffic.

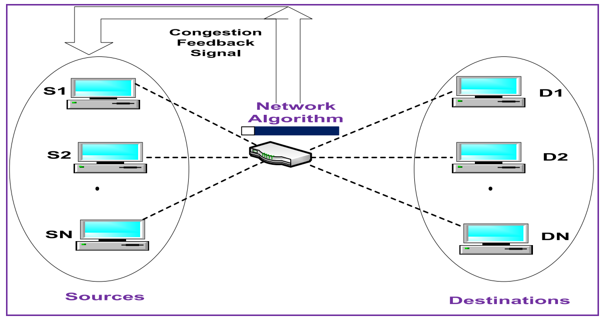

FIGURE 1. Network Model.

We adopted the model provided in [2]. We assumed the system is homogeneous for the sake of simplicity. The dynamic model is described by the Eqs.(7) -(8).

$$\dot{W}(t) = \frac{-2N}{R^2C} W(t) - \frac{RC^2}{2N^2} p(t-R) \quad (7)$$

$$\dot{q}(t) = \frac{N}{R} W(t) - \frac{q(t)}{R} \quad (8)$$

The parameters of Eq.(7) and Eq.(8) are shown in Table 2. The Eq.(7) represents the TCP window evolution where AIMD behavior is used during congestion control. For every round trip time the TCP window size is increased by one as shown in Eq.(7) as $[\frac{-2N}{R^2C} W(t)]$. Whenever a time out occurs in the TCP system, the window size decreased by half as shown in the second component of Eq.(7) as $[\frac{RC^2}{2N^2} p(t-R_0)]$. The Eq.(8)

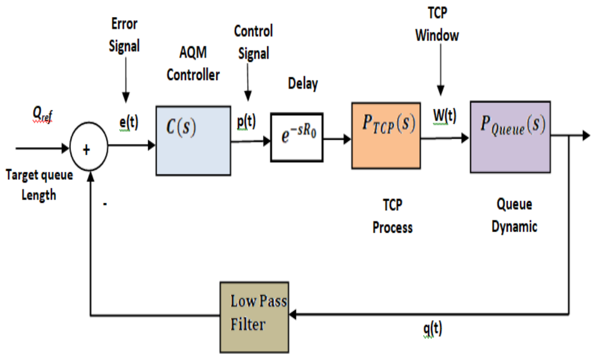

FIGURE 2. AQM design using control theory concepts.

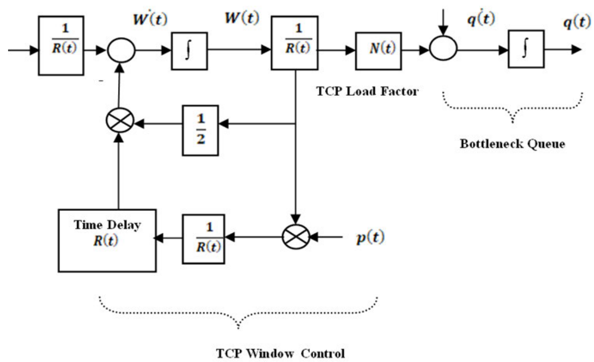

FIGURE 3. Mechanism of Congestion Avoidance and Flow Control in TCP system.

TABLE 2. Parameters of proposed model.

Parameters	Descriptions
W	TCP window size
$\dot{W}(t)$	Time-derivative of window size
N	Number of TCP sessions(Load factor)
C	Link capacity(Packets/sec)
R	Transmission round trip time
p	Packet dropping probability function (Positive number between [0, 1]
q	Current queue length(packets)
$\dot{q}(t)$	Time-derivation of queue length
Q_{ref}	Reference queue length(Target)

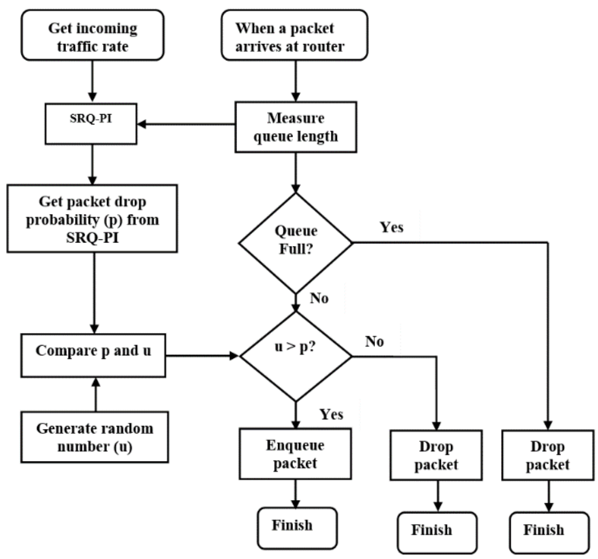

FIGURE 4. Simulation Design of SRQ-PI.

represents the queue dynamics during the packet transmission. The first component $[\frac{N}{R}W(t)]$ represents theincrease of queue length due to arrival of packets and the second component $[\frac{q(t)}{R}]$ represents decrease of queue length due to packet departure. The round trip time is measured using two component: propagation delay T_p and queuing delay $\frac{q(t)}{C}$ which is shown in Eq.(9).

$$R(t) = T_p + \frac{q(t)}{C} \tag{9}$$

The incoming traffic rate r(t)can be computed using Eq.(10).

$$r(t) = N\frac{W(t)}{R(t)} \tag{10}$$

In literature, different techniques have been proposed to measure the Internet traffic [37], [38] accurately. Author [39] proposed a new technique to predict change in arrival of packet and defined an appropriate dropping probability function based on the predictions. By considering Eqs.(7)-(9), the model is demonstrated in Fig. 3 which includes queue dynamics with window control system of TCP.

IV. THE PROPOSED CONTROLLER: SRQ-PI

Here, we describe the SRQ-PI algorithm and provide basic guidelines to select all the control gain parameter theoretically to achieve stability in the system with better performance. The SRQ-PI new dropping probability function is defined considering both incoming traffic rate and current queue length to regulate the incoming traffic to the capacity of the link and current queue length to the expected target value. The proposed algorithm stabilizes the queue length around the desired target and achieves faster transient response in the scenario where network conditions and parameters keep changing over time. The simulation design of SRQ-PI is demonstrated in Fig. 4.

A. NEW CONGESTION TRACKING FUNCTION

In order to achieve stability in queue length, the AQM controller needs to introduce new appropriate packet dropping probability function which can achieve faster convergence and reduce average jitter. The SRQ-PI defines new congestion control function to efficiently manage and control queue length around desired target and gets stability. The new congestion control function maps the levels of congestion based on incoming rate and queue length measured from each sampling time to the packet drop probability for the packets come towards the router. The new packet dropping function can be defined as in Eq.(11).

$$p(t) = p(t-1) + K_R(r(t) - C) + K_P\left(q(t) - Q_{ref}\right) - K_I\left(q(t-1) - Q_{ref}\right); \tag{11}$$

where K_R, K_I and K_P and are the rate constant, proportional and integral constant respectively. The packet drop probability function uses incoming traffic rate, queue length error and past queue length error. It considers past history (due to presence of integral part) to estimate the packet drop probability which make the controller insensitive to the sudden arrival TCP traffic and works better at multiple bottleneck link topology. The goal of SRQ-PI is to adopt dropping probability function to minimize the queue length error and to match the incoming traffic rate with the outgoing link capacity. We provide the control theoretic stability analysis and give systematic guidelines to choose control parameters.

In the design of SRQ-PI controller, we emphasize two things. The first one includes defining packet dropping probability function $p(t)$ and the second one is selecting values of K_R, K_P and K_I based on the guidelines. The new dropping probability functions: $p(t)$ computes two elements. First one includes error in incoming traffic (i.e. difference of incoming traffic rate and outgoing link capacity) and the second one includes queue length error (i.e. difference of current queue length and target). To achieve stability in the proposed system, selecting the proper value for control parameter is very crucial. So there is a need of proper analyze of system stability and selection parameter. Next, section we present the analysis of system stability and proper guidelines for selection of parameter.

B. STABILITY ANALYSIS

Using classical control theory, AQM controller can be analyzed with the stability, transient response and steady-state error. Stability measures the queue length oscillation around the given reference input (target). The stable system can increase the link utilization and reduce the queuing delay. The transient response (i.e. settling time) defines the time required for the queue length to reach to the given reference input. Sluggish transient response creates a barrier in maintaining queue length (controlled variable) around given reference input, if the network parameter change from time to time [10]. Steady state error measures the degree of the error signal. Keeping above measures in mind, the stability of the proposed system is analyzed based on control theory. The stability of SRQ-PI is measured based on the queue length fluctuations from time to time. We use Routh-Hurwitz theorem [35] of control theory to analyze the stability and provide guidelines to select K_R, K_P and K_I. Our proposed controller is designed based on control theory and shown in Fig. 5. As shown in figure the queue length (q) is used as internal feedback without needing any external inputs. The proposed controller stabilizes the system and adjusts the packet drop probability dynamically. At equilibrium point, SRQ-PI matches the queue length and the rate of incoming traffic to the target (Q_{ref}) and link capacity (C) respectively. The input rate is $r_o = N\frac{W_0}{R_0} = C$ and queue length is $q_o = Q_{ref}$.

Next, characteristic equation is obtained to analyze the stability of the proposed system which includes stable range of control parameter. By performing Laplace transforms, we linearize the network system and obtain the characteristic equation. Performing Laplace transforms of Eq.(7), Eq.(8) and Eq.(11), we obtain:

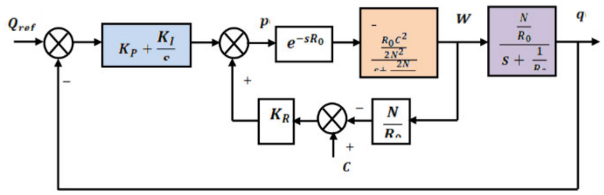

FIGURE 5. **SRQ-PI design using classical control theory.**

Step 1: Measure the current queue length;
Step 2: Compute the range of K_R by using Eqs.(34);
Step 3: Choose the suitable value of K_R within provided range;
Step 4: Compute the value of K_P and K_I using Eqs.(30), (38), (39) and (28);
Step 5: Choose the proper value of K_P and K_I within the range;
Step 6: Compute the packet drop probability (p) using Eqs.(11);
Step 7: To compute for next time slot (n+1) save the value of p and q;

FIGURE 6. **The SRQ-PI algorithm.**

$$sW(s) = W(0) - \frac{2N}{R^2C}W(s) - \frac{RC^2}{2N^2}P(s)e^{-sR} \tag{12}$$

$$sQ(s) = q(0) + \frac{N}{R}W(s) - \frac{Q(s)}{R} \tag{13}$$

$$\begin{aligned} sP(s) &= p(0) + K_R\left(\frac{N}{R}W(s) - \frac{C}{s}\right) \\ &\quad + K_P\left(Q(s) - \frac{Q_{ref}}{s}\right) - \frac{K_I}{s}\left(Q(s)e^{-sR} - \frac{Q_{ref}}{s}\right) \end{aligned} \tag{14}$$

The Laplace transforms of $W(t), q(t)$ and $p(t)$ is $W(s), Q(s)$ and $P(s)$ respectively. Taking Eqs.(11)-(13),characteristic function of Q(s) is obtained as:

$$Q(s) = \frac{B(s)}{A(s)} \tag{15}$$

where,

$$\begin{aligned} B(s) = &\left[\left(Rs + \frac{2N}{RC} + \frac{C^2R}{2Ns}e^{-sR}K_R\right)q(0) + NW(0)\right. \\ &- \frac{RC^2}{2Ns}e^{-sR}p(0) + \frac{RC^2}{2Ns^2}e^{-sR}Q_{ref}\left(K_P - \frac{K_I}{s}\right) \\ &\left. + \frac{RC^3e^{-sR}}{2Ns^2}K_R\right] \end{aligned} \tag{16}$$

$$\begin{aligned} A(s) = &(1 + sR)\left(s + \frac{2N}{R^2C} + \frac{C^2}{2Ns}e^{-sR}K_R\right) \\ &+ \frac{RC^2}{2N}e^{-sR}\left(\frac{K_P}{s} - \frac{K_I}{s^2}e^{-sR}\right) \end{aligned} \tag{17}$$

The characteristic equation $A(s)$ as given in Eq. (17) of the network system given is computed from the Eqs.(11)-(13). System stability can be analyzed through several methods. We adopted Routh-Hurwitz theorem [35] for determine the stability condition of the SRQ-PI and selection of control

TABLE 3. The Routh table.

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2	$\frac{a_2a_3 - a_1a_4}{a_3}$	a_0	0
s^1	$\frac{(((a_2a_3 - a_1a_4)/a_3)a_1) - a_3a_0}{(a_2a_3 - a_1a_4)/a_3}$	0	0

parameters. As stated in [40] the system achieves stable if all the zeros of $A(s)$ are in open left-half plane (OLHP). In order to compute the value for the characteristic equation $A(s)$, the second order approximation $e^{-sR} \approx 1 - sR + s^2R/2$ is used in the following analysis. Using second order approximation in Eq.(17), we have:

$$\begin{aligned} A(s) = s^4 &\left[R + \frac{C^2R^2}{4N}K_R - \frac{C^2R^3}{8N}K_I\right] \\ &+ s^3\left[1 + \frac{C^2R}{4N}K_R + \frac{2N}{RC} + \frac{C^2R^2}{4N}K_P - \frac{C^2R^2}{2N}K_R\right] \\ &+ s^2\left[\frac{2N}{R^2C} - \frac{C^2R^2}{2N}K_P - \frac{C^2R^2}{2N}K_I - \frac{C^2R^3}{2N}K_I\right] \\ &+ s^1\left[\frac{C^2}{2N}K_R + \frac{C^2R}{2N}K_P + \frac{C^2R^2}{N}K_I\right] \\ &+ s^0\left[\frac{C^2R}{2N}K_I\right] \end{aligned} \tag{18}$$

For the above equation (18), we let the coefficients of s^4, s^3, s^2, s^1 and s^0 are a_4, a_3, a_2, a_1 and a_0, respectively. We have:

$$a_4 = R + \frac{C^2R^2}{4N}K_R - \frac{C^2R^3}{8N}K_I \tag{19}$$

$$a_3 = 1 + \frac{C^2R}{4N}K_R + \frac{2N}{RC} + \frac{C^2R^2}{4N}K_P - \frac{C^2R^2}{2N}K_R, \tag{20}$$

$$a_2 = \frac{2N}{R^2C} - \frac{C^2R^2}{2N}K_P - \frac{C^2R^2}{2N}K_I - \frac{C^2R^3}{2N}K_I \tag{21}$$

$$a_1 = \frac{C^2}{2N}K_R + \frac{C^2R}{2N}K_P + \frac{C^2R^2}{N}K_I, \tag{22}$$

$$a_0 = \frac{C^2R}{2N}K_I \tag{23}$$

Considering the Eq.(18), the characteristic equation $A(s)$ is computed as:

$$A(s) = a_4s^4 + a_3s^3 + a_2s^2 + a_1s^1 + a_0s^0. \tag{24}$$

Then, Routh Table is prepared and shown in Table 3 for the network system.

For simplicity, let

$$b_1 = \frac{a_2a_3 - a_1a_4}{a_3}. \tag{25}$$

$$c_1 = \frac{(((a_2a_3 - a_1a_4)/a_3)a_1) - a_3a_0}{(a_2a_3 - a_1a_4)/a_3}. \tag{26}$$

As stated in Routh stability test, the system can be said as stable if the values computed in second column of Table 3 are positive, i.e.:

$$a_4 > 0, a_3 > 0, b_1 > 0, c_1 > 0, \text{ and } a_0 > 0. \tag{27}$$

Next, the elements of Eq.(27) is analyzed one by one for the stability condition of the system. Initially let us consider a_4 in Eq.(27). Since $N > 0$, $C > 0$, $R > 0$, to make $a_4 = R + \frac{C^2R^2}{4N}K_R - \frac{C^2R^3}{8N}K_I > 0$, we have:

$$K_I < \left(\frac{C^2R^2}{4N}K_R + R\right) \Big/ \left(\frac{C^2R^3}{8N}\right). \tag{28}$$

To make $a_3 > 0$, for the coefficient a_3 in Eq.(27) we have:

$$1 + \frac{C^2R}{4N}K_R + \frac{2N}{RC} + \frac{C^2R^2}{4N}K_P - \frac{C^2R^2}{2N}K_R > 0. \tag{29}$$

After solving the above equation, we have:

$$K_P > \left(\frac{C^2R^2}{2N}K_R - 1 - \frac{C^2R}{4N}K_R - \frac{2N}{RC}\right) \Big/ \left(\frac{C^2R^2}{4N}\right). \tag{30}$$

For b_1 in Eq. (25), we have:

$$\frac{a_2a_3 - a_1a_4}{a_3} > 0$$

Since the value of $a_3 > 0$, we obtained:

$$f(K_R) = a_2a_3 - a_1a_4 > 0. \tag{31}$$

The Eq. (31) can be expressed as a function of K_R. After replacing all the values of a_4, a_3, a_2, a_1 in the above Eq. (31), we have:

$$\begin{aligned} &\left[\frac{2N}{R^2C} - \frac{C^2R^2}{2N}K_P - \frac{C^2R^2}{2N}K_I - \frac{C^2R^3}{2N}K_I\right]\left[1 + \frac{C^2R}{4N}K_R \right. \\ &\quad \left. + \frac{2N}{RC} + \frac{C^2R^2}{4N}K_P - \frac{C^2R^2}{2N}K_R\right] \\ &\quad - \left[\frac{C^2}{2N}K_R + \frac{C^2R}{2N}K_P + \frac{C^2R^2}{N}K_I\right]\left[R + \frac{C^2R^2}{4N}K_R \right. \\ &\quad \left. - \frac{C^2R^3}{8N}K_I\right] > 0. \end{aligned} \tag{32}$$

It is found that computed Eq.(32) is a quadratic function of K_R. Now we get the coefficient of K_R^2 as: $-\left[\frac{C^4R^2}{8N^2}\right]$

Since $N > 0$, $C > 0$ and $R > 0$, the coefficient of K_R^2 is

$$-\left[\frac{C^4R^2}{8N^2}\right] < 0. \tag{33}$$

As the coefficient of K_R^2 is non-positive, the function $f(K_R)$ in Eq.(31) is the parabolic curve and arises two points of K_R which make $K_R^2 = 0$. Assume the two points are K_a and K_b that make $f(K_R) = 0$ and $K_a < K_b$. In order to make the Eq.(31) true, the range of K_R must be:

$$K_a < K_R < K_b. \tag{34}$$

For c_1 in Eq.(26), we have:

$$\frac{(((a_2a_3 - a_1a_4)/a_3)a_1) - a_3a_0}{(a_2a_3 - a_1a_4)/a_3} > 0.$$

Since $(a_2a_3 - a_1a_4)/a_3 > 0$, we obtained:

$$f(K_P) = a_1a_2a_3 - a_1^2a_4 - a_3^2a_0 > 0. \tag{35}$$

In order to compute the value of range of K_P, the equation (35) can be defined as a function of K_P. As the value of a_4 and a_0, are not dependant on K_P, we substitute the value of a_1, a_2 and a_3 in Eq.(35) and we get:

$$\left[\frac{C^2}{2N}K_R + \frac{C^2R}{2N}K_P + \frac{C^2R^2}{N}K_I\right]\left[\frac{2N}{R^2C} - \frac{C^2R^2}{2N}K_P - \frac{C^2R^2}{2N}K_I - \frac{C^2R^3}{2N}K_I\right]\left[1 + \frac{C^2R}{4N}K_R + \frac{2N}{RC} + \frac{C^2R^2}{4N}K_P - \frac{C^2R^2}{2N}K_R\right] - \left[\frac{C^2}{2N}K_R + \frac{C^2R}{2N}K_P + \frac{C^2R^2}{N}K_I\right]^2 a_4 - \left[1 + \frac{C^2R}{4N}K_R + \frac{2N}{RC} + \frac{C^2R^2}{4N}K_P - \frac{C^2R^2}{2N}K_R\right]^2 a_0 > 0 \tag{36}$$

The Eq.(35) is a cubic function of K_P. Now we obtained coefficient of K_P^3 as: $-\left[\frac{C^6R^5}{16N^3}\right]$

Since $N > 0$, $C > 0$ and R > 0, the coefficient of K_P^3 is

$$-\left[\frac{C^6R^5}{16N^3}\right] < 0. \tag{37}$$

The coefficient of K_P^3 is negative. Since the function $f(K_P)$ in Eq.(35) is cubic function, there exist three points of K_P which make $f(K_P) = 0$. Let the three points be K_1, K_2 and K_3 that makes $f(K_P) = 0$ and $K_1 < K_2 < K_3$. In order to make the Eq.(35) true the range of K_P must be:

$$K_1 < K_P < K_2. \tag{38}$$

or

$$K_2 < K_P < K_3. \tag{39}$$

Considering the last value of the second column (a_0), in Table 2, we have:
$\frac{C^2R}{2N}K_I > 0$. Since $N > 0$, $C > 0$ and $R > 0$, we have:

$$K_I > 0. \tag{40}$$

From the above analysis the stability condition of control gain K_R determined from Eq.(34), and K_P from Eq.(30), Eq.(38) and Eq.(39) and K_I from Eq.(28). The range provides guidelines to select the parameter of SRQ-PI. However, this analysis does not provide optimal value of control gain parameter. These parameters can be obtained through simulation analysis or in real scenario. To estimate the parameters N, C and R many methods have been proposed in past literature. In [19], [41] proposed a method to estimate the parameters such as N and C.

Network parameters such as traffic load (N), Bottleneck capacity (C) and round trip time (R) dependent on time-varying nature. Based on dynamic behavior of congestion window length of TCP, the traffic load is calculated as given in Eq.(41).

$$N = r(t)\frac{R(t)}{W(t)} \tag{41}$$

TABLE 4. The Routh table for example 1.

s^4	0.0642	60.636	10
s^3	6.521	14.725	0
s^2	60.491	10	0
s^1	13.647	0	0
s^0	10	0	0

The bottleneck link capacity is estimated by measuring incoming traffic rate, $r(t)$. To estimate the bottleneck capacity Eq.(42) is used. The bottleneck link capacity can also be estimated using spectral and statistical analysis of traffic.

$$C = \frac{q(t)}{R(t) - T_p} \tag{42}$$

The round trip time is measured using two component: propagation delay $\mathrm{T_p}$ and queuing delay $\frac{\mathrm{q(t)}}{\mathrm{C}}$ which is shown in Eq.(9).

As indicated properly our systematic analysis gives a clear relationship among network parameters and stability condition of the system as done in the other work [36], [42].

Example 1: For a network system with parameters: $N = 250$, $C = 1250$ packets/sec, $R = 0.08$ sec, $K_R = 0.0002$, $K_P = 0.05$ and $K_I = 0.04$ we discuss the stability of the system.

From Eqs.(19)-(23), we have $a_4 = 0.0642$, $a_3 = 6.521$, $a_2 = 60.636$, $a_1 = 14.725$ and $a_0 = 10$. The values of Routh table is computed and shown in Table 4. For the system stability, all the elements present in the second column of Table 4 must be positive.

Since, all the values obtained in the second column of Table 4 is greater than zero (positive), the system is stable. Alternatively, we test the stability of the system using the MATLAB for Routh-Hurwitz test and obtained the same results as shown in Table 3.

C. THE SRQ-PI ALGORITHM

In this section we provide the detail of the proposed algorithm in Fig. 6. The control gain parameters of SRQ-PI can be selected based on guidelines provided in the proposed algorithm. The value of K_R, K_P and K_I is selected after computing the range for each control gain parameter. We have not provided any theoretical background to select the best value for the control gain at this moment. We have validated our proposed SRQ-PI algorithm by implementing in NS2. The system achieves better stability having the lower fluctuation in queue length. The values for K_R, K_P and K_I are selected within the range to make the system stable and achieve faster transient response and to reduce the average delay jitter.

D. FERFORMANCE EVALUATION OF SRQ-PI

Here, performance of SRQ-PI algorithm is analyzed and compared it with PI, IAPI [43], and REM through popular network simulator, NS2 [44]. We consider a single bottleneck topology as depicted in Fig. 7. The bandwidth of side link is set to 20Mb and delay to 10ms. The bottleneck link capacity (C) is 10Mb (1250 packets/sec) and propagation delay

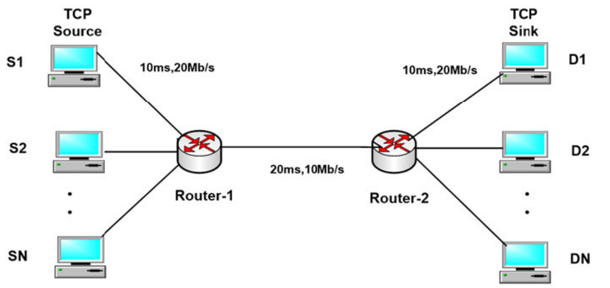

FIGURE 7. A single bottleneck topology.

TABLE 5. Summary of parameters for AQM controller.

AQM Controller	Parameters
SRQ-PI	$K_P = 5 * 10^{-2}$ and $K_I = 4 * 10^{-2}$, $K_R = 2 * 10^{-4}$, $Q_{ref} = 200$, ω =170(sampling frequency)
IAPI	$k_p = 0.00014723, k_{i0} = 0.0000004277, e^* =$ 40 and $k_1 = 0.5$, and $\omega = 170$(sampling frequency)
PI	$a =$ 0.00001822, $b = 0.00001816$ and ω =170(sampling frequency)
REM	$pbo\_$ =200, $\gamma = 0.001$ and $\phi = 1.001$

is 20ms. There are 250 (N) number of TCP connections are created in the networks having equal round trip time (R_0 = 0.08 sec) of each where size of each packet is 1000 bytes. The limit of buffer size is considered to 600 packets. In order to manage the queue efficiently we used SRQ-PI in router R1 and Droptail in router R2. All the sources use TCP/Reno and generate traffic at the same time (0sec). Each simulation runs for 100 sec. It is assumed that some parameters have default values if it does not have any explicit value. The target queue length (Q_{ref}) is 200 packets. The parameters used for the different AQM controller is given in Table 5.

Here, simulation experiment results are provided using the topology shown in Fig. 7. Then stability of the proposed scheme is analyzed using a current queue length. Here the settling time means time required for the controller to maintain the current queue length into a steady state.

E. PERFORMACNE ANALYSIS WITH EXISTING AQM CONTROLLERS

Initially, performance of SRQ-PI is compared with IAPI, REM and PI using queue length (packets), average delay (ms), average jitter, packet drop rate and mean queue length and standard deviation. We set the target queue length at 200 packets. The results are shown in Fig.8 and Table 6 for 250 TCP connections. We can see the performance of SRQ-PI is better than existing AQM schemes in link utilization, average delay, average jitter, settling time, mean and std.dev. of queue length. SRQ-PI achieves lower average delay and average jitter than others. The controller getting lower delay jitter is suitable for the multimedia application which is very sensitive to delay jitter [45]. The SRQ-PI is more efficient in controlling the current queue length around the given desired target as its mean queue length is 195.78 which is much close to the 200 (target).

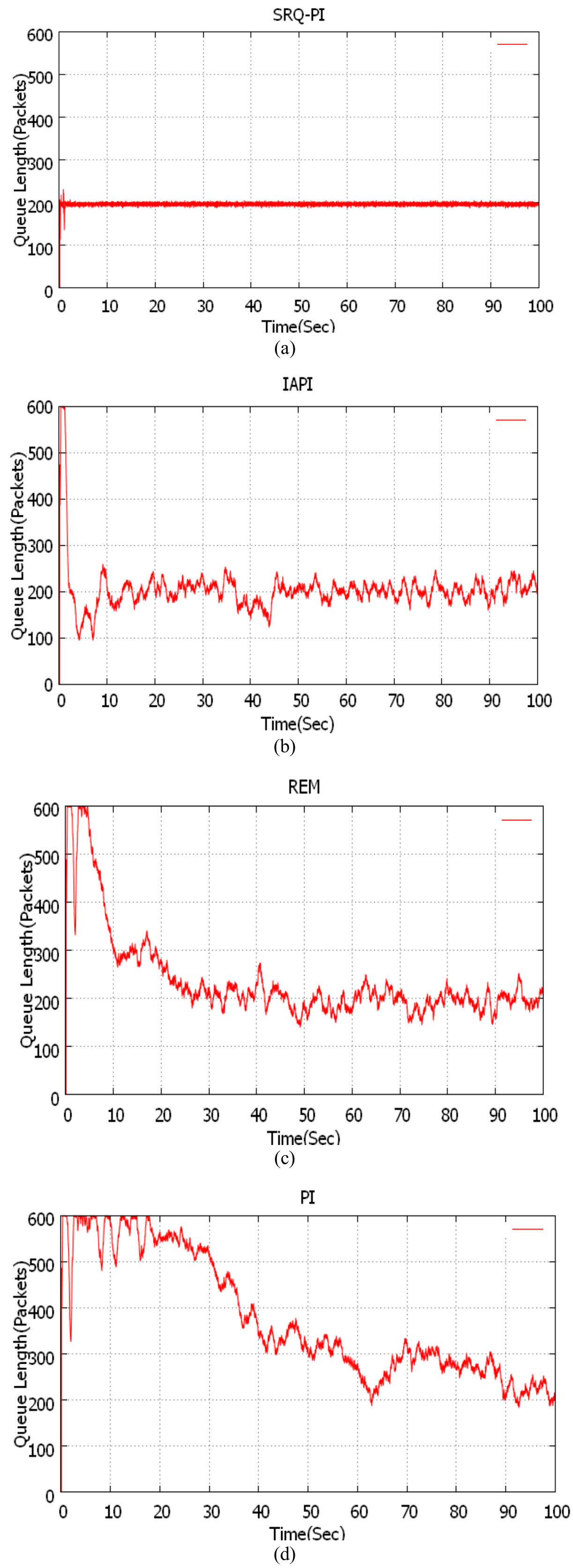

FIGURE 8. Measured Queue length of (a) SRQ-PI (b) IAPI(c) REM (d) PI.

The proposed controller, SRQ-PI can efficiently control the queue length with small overshoot up to 2 sec and

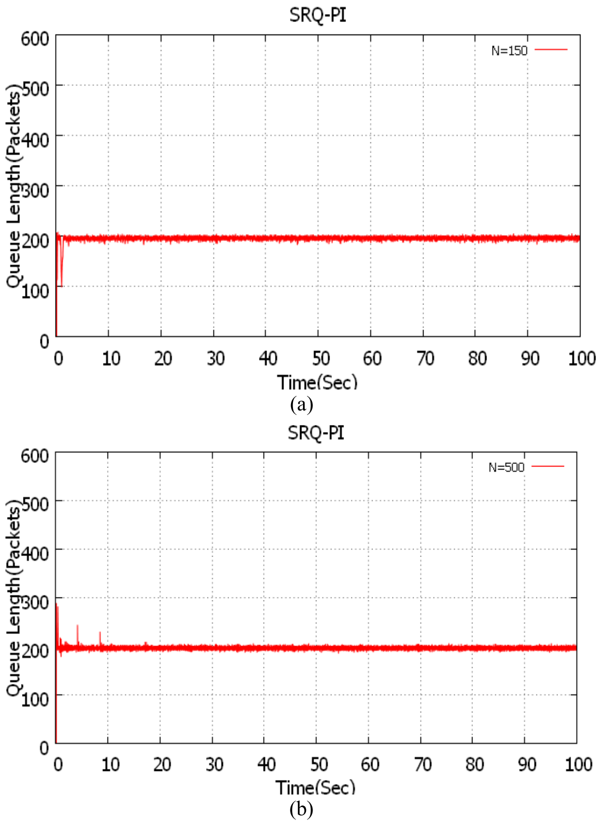

FIGURE 9. Measured Queue length of SRQ-PI for TCP connections (a) N = 150 (b) N = 500.

TABLE 6. Comparison of results of AQM controllers.

	SRQ-PI	IAPI	REM	PI
Link utilization(%)	**96.047**	96.047	96.047	96.047
Average Delay(ms)	**0.122047**	0.125427	0.139690	0.197169
Average Jitter	**0.000547**	0.002192	0.003767	0.004531
Mean(Packets)	**195.78**	204.47	237.42	374.40
Settling time(Sec)	**1.8745**	10.4789	30.1245	90.3464
Packet drop rate(%)	**20.14**	19.55	18.26	14.31
Std.dev.(Packets)	**56.1177**	56.1916	56.4344	56.6413

then manages queue efficiently until end of the simulation as shown in Fig.8 (a). The time required to bring the queue length(settling time) into the target(200 packet) is less than 2 sec which is very low as compared to REM, RED and PI as shown in Fig.8. On the other way, the settling time of SRQ-PI is much faster than REM, RED and PI(Refer Table 6) due to proper coupling between queue length and marking probability in SRQ-PI.

F. PERFORMANCE UNDER DIFFERENT TCP CONNECTION

The performance of SRQ-PI is tested with a variable number of TCP connections. Here, the queue length for 150 and 500 TCP connections (N) is measured and shown in Fig. 9 shows. We can see that, the SRQ-PI efficiently controls the queue length and tries to manage at desired target with faster setting time under variation of TCP load. As shown in Fig.9 for light TCP connection(N = 150) the oscillation is observed at the beginning of the simulation up to 2 sec and then tries to manage efficiently till the end of the simulation. However, heavy traffic load (N = 500) brings more oscillation up to 9 sec of the simulation due to the rapid variation of the states of the network.

G. PERFORMANCE UNDER DIFFERENT LINK CAPACITY

Here, performance of SRQ-PI is analyzed under different bottleneck link capacity(C). The capacity of the bottleneck link is set with two different values: 2Mb and 50Mb. The queue length for 250 TCP connections is depicted in Fig. 10. It is marked from the figure that SRQ-PI efficiently manage the queue length with small oscillation around the operating point under different link capacity. However, the oscillation increases with increase of bottleneck capacity because the inertia of the system increases.

H. PERFORMANCE UNDER DIFFERENT ROUND TRIP TIMES

In this section, performance of SRQ-PI is tested for different round trip (RTT) times of TCP connections. In order to do so the RTT (τ) of each connection is set to 50ms for the first run and 120ms for the second run. Keeping 250 TCP connections the queue length is shown in Fig. 11. It is observed from the figure that the proposed controller efficiently controls the queue length around target with small oscillation. It required few seconds to reach the steady state under different RTT which signifies that the SRQ-PI is more efficient in stabilizing the queue length around given target under different RTTs.

Next, performance of SRQ-PI is analyzed for different group of TCP connections. Each group of 100 TCP connections has different round trip times. The first group has the RTT 80ms and the second group has RTT 120ms and the third group has RTT 160ms. The queue length of SRQ-PI is shown in Fig. 12. We can clearly observe that SRQ-PI efficiently controls the queue length with small oscillations. The queue length of SRQ-PI converges to the target quickly. The proposed controller maintains the same stability among the three methods under any given value of RTT. The SRQ-PI controller is less sensitive to the RTT variation with different group of TCP connection and regulates the queue length to the operating point of 200 packets.

I. PERFORMANCE UNDER SUDDEN CHANGE OF TRAFFIC LOADS

Here, the sensitivity to the sudden change of TCP traffic loads of SRQ-PI is analyzed. In order to analyze sensitivity, we add and drop traffic loads abruptly during simulation time. Here we study time-varying dynamics during simulation. At the beginning of the simulation experiment, 250 TCP connections are generated. Additional 100 TCP connections generated at 50 Sec and dropped at 70 sec. The queue length of the proposed controller is shown in Fig. 13. The proposed SRQ-PI is efficient enough in stabilize queue length

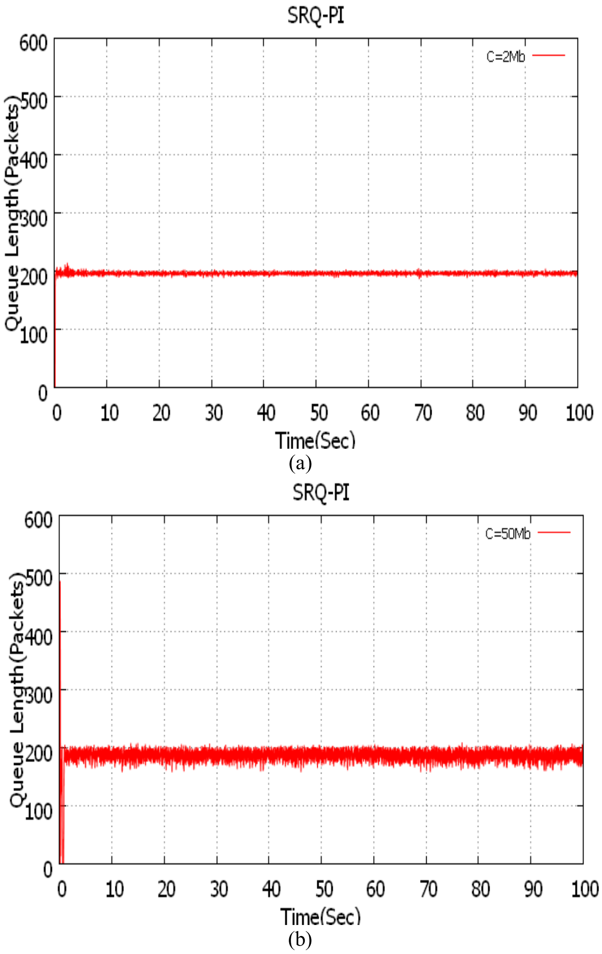

FIGURE 10. Measured Queue length of SRQ-PI with bottleneck bandwidth (a) C = 2Mb (b) C = 50Mb.

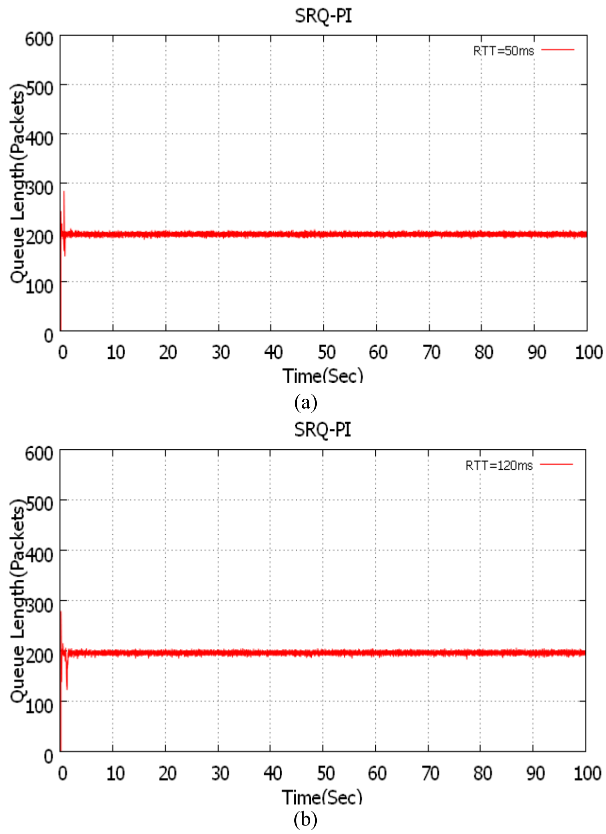

FIGURE 11. Measured Queue length of SRQ-PI with RTT (a) $\tau = 50$ms (b) $\tau = 120$ms.

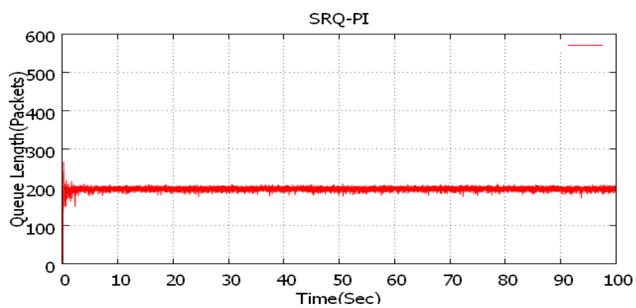

FIGURE 12. Measured Queue length of SRQ-PI with different groups of TCP connection with variation of RTTs.

around given target for different time varying TCP traffic. The response time of SRQ-PI is very fast under the sudden change of traffic loads. SRQ-PI controller performs better, responding more quickly to load variations. After 70 sec the controller exhibit more responsive behavior up to end of the simulation. The proposed controller performs better, responding more quickly to load variations. We see that the system response is stable, with fluctuations about an operating level of the queue. It shows that the SRQ-PI is robust and able to maintain stability under time varying TCP traffic as compared to others.

J. PERFORMANCE UNDER RANDOM START TIME

In this, performance of SRQ-PI for 250 TCP connections is analyzed involving random start time. We performed three simulations experiments for three different scenarios. In the first simulation, the starting time of each TCP connection is random between 0 to 10 sec and ending time is constant at 100 sec. For these two scenarios the queue length of SRQ-PI is measured and shown in Fig. 14, Fig. 15 and Fig. 16. We can observe that SRQ-PI is able to control the queue length around target and achieve faster convergence time. It took few seconds to reach in steady state which demonstrates that SRQ-PI stabilizes the queue length around the target efficiently. It shows that SRQ-PI controller adapts itself to the variations of the random start time to regulate queue length accurately from time to time.

K. EFFECT OF SRQ-PI ON MULTIPLE BOTTLENECK

In all above, performance of proposed technique is analyzed under single bottleneck topology. Here, we analyzed the impact of multiple bottleneck links as done in [46] on the performance of proposed technique as shown in Fig. 17. As the name indicates the multiple bottleneck topology, will have two bottleneck links. One is from router R2 to R3 and the other bottleneck link from router R4 to R5. The delay and

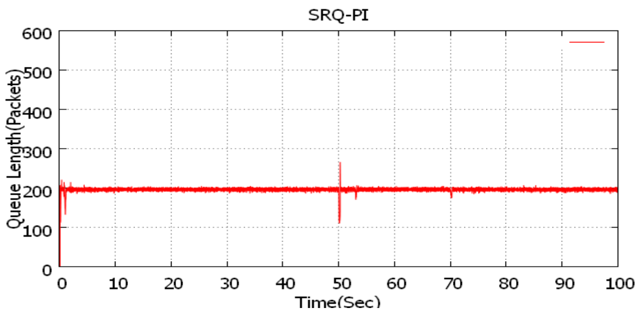

FIGURE 13. Measured Queue length of SRQ-PI with sudden change of traffic.

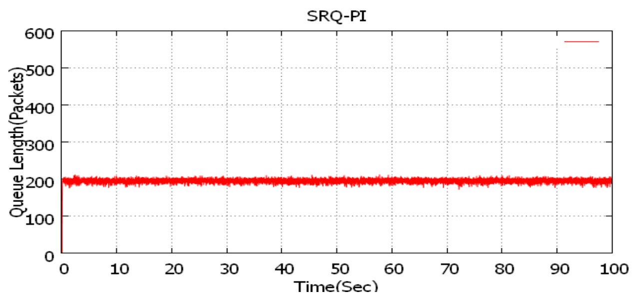

FIGURE 14. Measured Queue length of SRQ-PI where starting time is random.

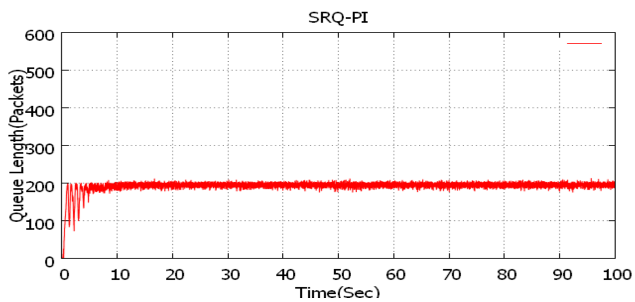

FIGURE 15. Measured Queue length of SRQ-PI under random RTT and random start time.

bandwidth between routers is 20ms and 10Mb respectively. The delay and capacity of other links are 10ms and 20Mb, respectively. In this topology, three different sets of TCP traffic are created named as set-1, set-2 and set-3. In set-1 group, we have created 250 TCP connections, where as 50 TCP connections are created in set-2 and set-3 each. The RTT for each TCP present in set-1 is 240ms and RTT of 80ms in each set-2 and set-3. The 250 TCP connections created in set-1 are passes through all router R1 and router R6. The 50 TCP connections created in each set-2 is passes through router R2 and router R3 and another 50 TCP connections in set-3 passes through router R4 and router R5.

The queue length of SRQ-PI using multiple bottleneck link is shown in Fig. 18. We can see the SRQ-PI able to stabilize the queue length around the target at both the routers R2 and R4. Moreover, it is also found that router R2 is less efficient than R4 in stabilizing the queue length.

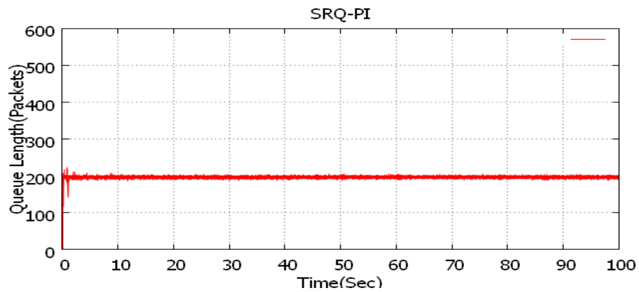

FIGURE 16. Measured Queue length of SRQ-PI with the sudden arrival of TCP load with random start time.

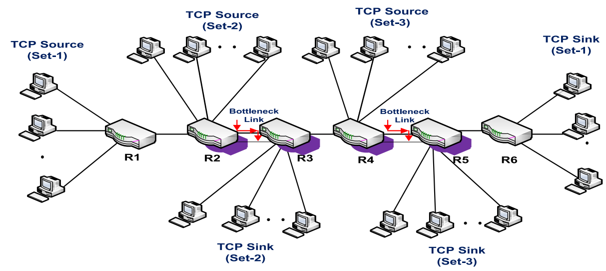

FIGURE 17. The multiple bottleneck network topology.

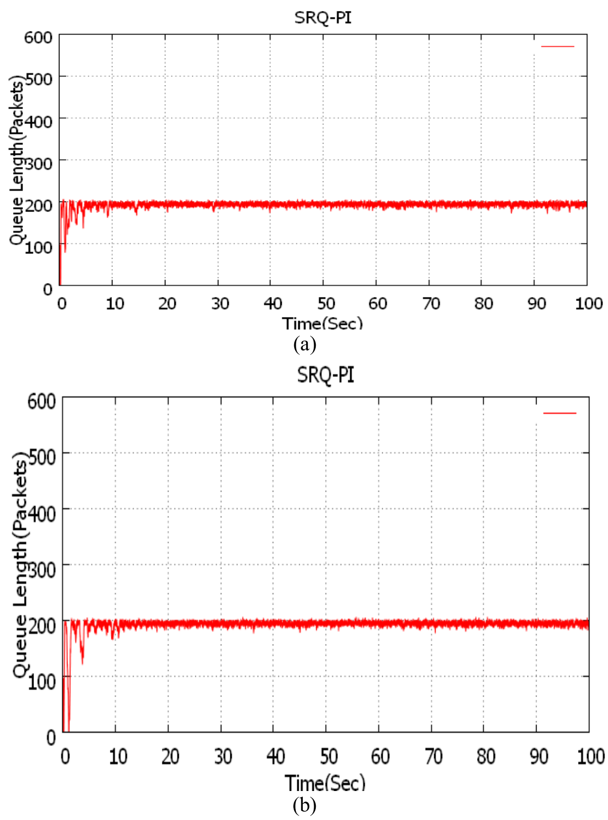

FIGURE 18. Measured Queue length of SRQ-PI for Router (a) R2 (b) R4.

L. EFFECT OF UNRESPONSIVE FLOWS ON THE PERFORMANCE OF SRQ-PI

In above all cases we have considered uniform responsive flow (TCP) is considered without considering any UDP disturbances. Here, presence of unresponsive flow (UDP) on the performance of SRQ-PI is analyzed. In order to do so

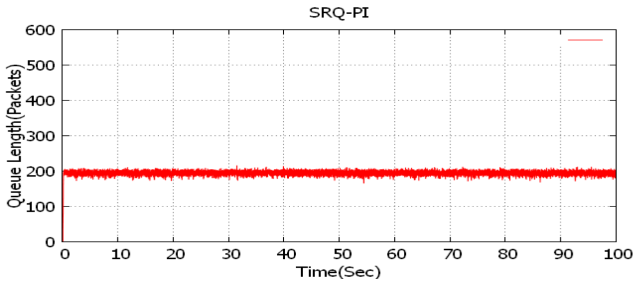

FIGURE 19. **Queue length of SRQ-PI for mixed traffic of 250 TCP and 50 UDP.**

we have mixed 250 TCP connections with 50 UDP flows where RTT of each TCP connection is uniformly distributed between 60 and 160ms. Each UDP flow will have uniform distribution of propagation delay between 30 and 150ms and follows exponential ON/OFF traffic model. The idle and burst time has a mean of 0.5 sec each. The rate of sending during on time would be 64kb/s having size of packet of 210 bytes. Then current queue length dynamics is measured for SRQ-PI and shown in Fig. 19. We can clearly see the SRQ-PI effectively stabilizes the queue length and robust under UDP disturbances. The convergence time of SRQ-PI is faster with small oscillations even when UDP flow mixed with TCP connections.

V. CONCLUSION

A self-tuning feedback controller is proposed measuring both queue length and input traffic rate to achieve stability in the network system. For stability analysis, the Routh-Hurwitz theorem is used. It provides the proper guidelines to select gain parameters to stabilize the system. Then performance of proposed SRQ-PI is analyzed under dynamic scenarios where the TCP connections, bandwidth and RTT keep changing. In addition, the performance of SRQ-PI is analyzed under the sudden change of traffic loads, TCP connection mixed with UDP flow and using multiple bottleneck links. Finally, SRQ-PI is compared with REM, IAPI, and PI.

The SRQ-PI efficiently controls the queue length with small overshoot and achieves better settling time as compared to REM, IAPI and PI. The mean queue length of proposed controller is lower and close to the target. The proposed controller is not sensitive to the sudden change of traffic loads and performs better at multiple bottleneck link. The SRQ-PI achieves lower average delay, lower average jitter and lower mean and standard deviation than other existing AQM schemes.

REFERENCES

[1] B. Braden, *Recommendations on Queue Management and Congestion Avoidance in the Internet*, document RFC 2309, 1998.
[2] V. Misra, W.-B. Gong, and D. Towsley, "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.
[3] R. Adams, "Active queue management: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1425–1476, 3rd Quart., 2013.
[4] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue (AVQ) controller for active queue management," in *Proc. ACM SIGCOMM*, New York, NY, USA, 2001, pp. 123–134.
[5] W. Feng, K. Shin, D. D. Kandlur, and D. Saha, "The blue active queue management controller," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 513–528, 2002.
[6] B. Wydrowski and M. Zukerman, "GREEN: An active queue management controller for a self managed Internet," in *Proc ICC*, 2002, pp. 2368–2372.
[7] 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.
[8] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A control theoretic analysis of RED," in *Proc. IEEE Conf. Comput. Commun. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Apr. 2001, pp. 1510–1519.
[9] J. Sun, G. Chen, K.-T. Ko, S. Chan, and M. Zukerman, "PD-controller: A new active queue management scheme," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, 2003, pp. 3103–3107.
[10] F. Y. Fie, R. Fengyuan, and L. Chuang, "Design a PID controllers for active queue management," in *Proc. 8th IEEE Symp. Comput. Commun.*, 2003, pp. 985–990.
[11] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, "REM: Active queue management," *IEEE Netw.*, vol. 15, no. 3, pp. 48–53, 2001.
[12] J. Sun and M. Zukerman, "RaQ: A robust active queue management scheme based on rate and queue length," *Comput. Commun.*, vol. 30, no. 8, pp. 1731–1741, Jun. 2007.
[13] E.-C. Park, H. Lim, K.-J. Park, and C.-H. Choi, "Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks," *Comput. Netw.*, vol. 44, no. 1, pp. 17–41, Jan. 2004.
[14] C. Long, X. Guan, B. Zhao, and J. Yang, "The yellow active queue management controllers," *Comput. Netw.*, vol. 47, no. 4, pp. 550–525, 2005.
[15] S. Floyd, R. Gummadi, and S. Shenker. (2001). *Adaptive RED: A Controllers for Increasing the Robustness of RED's Active Queue Management*. [Online]. Available: http://www.Icir.org/floyd/papers/adaptiveRed.pdf
[16] J. Sun, K.-T. Ko, G. Chen, S. Chan, and M. Zukerman, "PD-RED: To improve the performance of RED," *IEEE Commun. Lett.*, vol. 7, no. 8, pp. 406–408, Aug. 2003.
[17] T. H. Kim and K. H. Lee, "Refined adaptive RED in TCP/IP networks," in *Proc. SICE-ICASE*, 2006, pp. 3722–3725.
[18] K. Zhou, K. L. Yeung, and V. O. K. Li, "Nonlinear RED: A simple yet efficient active queue management scheme," *Comput. Netw.*, vol. 50, no. 18, pp. 3784–3794, Dec. 2006.
[19] T. J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: Stabilized RED," in *Proc. IEEE Conf. Comput. Commun. 18th Annu. Joint Conf. IEEE Comput. Commun. Soc. Future Now (INFOCOM)*, Mar. 1999, pp. 1346–1355.
[20] M. P. Tahiliani, K. C. Shet, and T. G. Basavaraju, "CARED: Cautious adaptive RED gateways for TCP/IP networks," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 857–864, Mar. 2012.
[21] C. Wang, J. Liu, B. Li, K. Sohraby, and Y. T. Hou, "LRED: A robust and responsive AQM algorithm using packet loss ratio measurement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 1, pp. 29–43, Jan. 2007.
[22] S. Liu, T. Basar, and R. Srikant, "Exponential-RED: A stabilizing AQM scheme for low- and high-speed TCP protocols," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 1068–1081, Oct. 2005.
[23] J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Comput. Netw.*, vol. 36, nos. 2–3, pp. 203–235, Jul. 2001.
[24] L. Ma, X. Liu, H. Wang, and X. Deng, "Congestion tracking control for multi-router TCP/AQM network based on integral backstepping," *Comput. Netw.*, vol. 175, pp. 1–10, Apr. 2020, doi: 10.1016/j.comnet.2020.107278.
[25] R. Hotchi, H. Chibana, T. Iwai, and R. Kubo, "Active queue management supporting TCP flows using disturbance observer and smith predictor," *IEEE Access*, vol. 8, pp. 173401–173413, 2020.
[26] H. Abdel-Jaber, "An exponential active queue management method based on random early detection," *J. Comput. Netw. Commun.*, vol. 2020, pp. 1–11, May 2020.
[27] N. El Fezazi, "An observer-based controller for congestion control in data networks," *J. Control, Autom. Electr. Syst.*, vol. 30, no. 5, pp. 645–656, Oct. 2019.
[28] 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., IEICE*, vol. 10, no. 2, pp. 157–172, 2019, doi: 10.1587/nolta.10.157.

[29] S. Belamfedel Alaoui, E. H. Tissir, and N. Chaibi, "Active queue management based feedback control for TCP with successive delays in single and multiple bottleneck topology," *Comput. Commun.*, vol. 117, pp. 58–70, Feb. 2018, doi: 10.1016/j.comcom.2018.01.003.
[30] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, *Controlled Delay Active Queue Management*, document RFC 8289, Jan. 2018.
[31] P. Wang, C. Zhu, and X. Yang, "A novel AQM algorithm based on feedforward model predictive control," *Int. J. Commun. Syst.*, vol. 61, no. 12, p. e3711, 2018.
[32] 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.
[33] R. Shorten, F. Wirth, and D. Leith, "A positive systems model of TCP-like congestion control: Asymptotic results," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 16–29, Jun. 2006.
[34] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Comput. Chem. Eng.*, vol. 23, nos. 4–5, pp. 667–682, May 1999.
[35] V. Veselý, D. Rosinová, and M. Foltin, "Robust model predictive control design with input constraints," *ISA Trans.*, vol. 49, no. 1, pp. 114–120, Jan. 2010.
[36] Q. Xu, F. Li, J. Sun, and M. Zukerman, "A new TCP/AQM system analysis," *J. Netw. Comput. Appl.*, vol. 57, pp. 43–60, Nov. 2015.
[37] M. Celenk, T. Conley, J. Willis, and J. Graham, "Predictive network anomaly detection and visualization," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 288–299, Jun. 2010.
[38] M. X. Yan and L. H. Guangr, "A novel parameter tuning algorithm for AQM-PI controllers," *Res. J. Appl. Sci., Eng. Technol.*, vol. 4, no. 1, pp. 75–78, 2012.
[39] Q. Xu and J. Sun, "A simple active queue management based on the prediction of the packet arrival rate," *J. Netw. Comput. Appl.*, vol. 42, pp. 12–20, Jun. 2014.
[40] W. Kamen and B. S. Heck, *Fundamentals of Signals and Systems Using the Web and MATLAB*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[41] H. Zhang, D. Towsley, C. V. Hollot, and V. Misra, "A self-tuning structure for adaptation in TCP/AQM networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 302–303, Jun. 2003.
[42] L. Yuan, C.-N. Chuah, and P. Mohapatra, "ProgME: Towards programmable network MEasurement," *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 115–128, Feb. 2011.
[43] J. Sun, S. Chan, and M. Zukerman, "IAPI: An intelligent adaptive PI active queue management scheme," *Comput. Commun.*, vol. 35, no. 18, pp. 2281–2293, Nov. 2012.
[44] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, 1st ed. Cham, Switzerland: Springer, 2008.
[45] M. Claypool and T. Jonathan, "The effects of jitter on the perceptual quality of video," in *Proc. 7th ACM Int. Conf. Multimedia*, 1999, pp. 115–118.
[46] K. Matsushima, K. Hirata, and M. Yamamoto, "Fairness improvement of multiple-bottleneck flow in data center networks," *IEICE Trans. Commun.*, vol. E99.B, no. 7, pp. 1445–1454, 2016.

SUKANT KISHORO BISOY received the master's degree in computer science and engineering from Visvesvaraya Technological University (VTU), Belgaum, India, in 2003, and the Ph.D. degree from Siksha 'O' Anusandhan University, India, in 2017. He is currently working as an Associate Professor and the Head of the Department of Computer Science and Engineering, C.V. Raman Global University, India. He has been involved in organizing many conferences, workshop, and FDP. He has several publications in national and international conference and journals and given invited talk in many workshops. His current research interests include wireless sensor networks, neuro-robotics, machine learning, cloud computing, and SDN. He is a Reviewer of the IEEE, Elsevier, and Springer Journal and conference.

PRASANT KUMAR PATTNAIK (Senior Member, IEEE) is currently working as a Professor with the School of Computer Engineering, KIIT University, Bhubaneswar, India. He has more than a decade of teaching and research experience. He has published numbers of research papers in peer-reviewed international journals and conferences. His research interests include mobile computing, cloud computing, brain–computer interface, and privacy preservation.

MANGAL SAIN received the M.Sc. degree in computer application, India, in 2003, and the Ph.D. degree in computer science, in 2011. Since 2012, he has been an Assistant Professor with the Department of Computer Engineering, Dongseo University, South Korea. He has authored over 70 international publications, including journals and international conferences. His research interests include wireless sensor networks, cloud computing, the Internet of Things, embedded systems, and middleware. He is a member of TIIS and a TPC member of more than ten international conferences.

DO-UN JEONG received the B.S. degree in electronics engineering from Dongseo University, Busan, South Korea, in 2000, and the M.S. and Ph.D. degrees in biomedical engineering from Pusan National University, Busan, in 2002 and 2005, respectively. He is currently a Professor with the Department of Information and Communication Engineering, Dongseo University. His research interests include biomedical sensor, signal processing, and smart-health care.