



# A New Modified Dropping Function for Congested AQM Networks

Sanjeev Patel<sup>1,2</sup>  · Karmeshu<sup>2</sup>

Published online: 29 September 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

## Abstract

Active queue management schemes are used to reduce the number of dropped packets at the routers. Random early detection uses dropping probability which is calculated based on the average queue size. Further it is modified according to the value of the *count* indicating the number of unmarked packets that have arrived after a marked packet. The impact of random variable i.e. number of packets arrived after a marked packet over the dropping pattern is investigated. The proposed model achieves smooth dropping pattern which results in improvement of quality of service parameters. A new model for dropping probability is proposed with different dropping function. The effect of new dropping probability results in the increase of the throughput and reduction of the expected end-to-end delay. An important finding is that the choice of modified dropping function significantly affects the performance measures of the networks.

**Keywords** Active queue management · Random early detection · Modified dropping probability · End-to-end delay · Throughput · Loss-rate · Traffic load · Buffer size

## 1 Introduction

Active queue management (AQM) is one of the popular network assisted congestion control algorithms which is implemented at the routers/gateways. The AQM scheme may involve one or more parameters based on queue size, loss-rate, input–output rate mismatch and queue mismatch, delay, or based on controller [1–7]. Random early detection (RED) [1], Adaptive-RED [2], triple-section RED (TRED) [6] are well known queue size based AQM schemes. The dropping probability in RED, TRED, and Adaptive-RED is computed using the average queue size,  $min_{th}$ ,  $max_{th}$ , and  $max_p$  and it is further modified to incorporate the number of unmarked packets to have arrived since a last marked packet.

The throughput and average queuing delay are concerned, RED is sensitive both to the traffic load as well as parameter settings [2]. The RED has limitation to keep

---

✉ Sanjeev Patel  
patelcs01@gmail.com

<sup>1</sup> Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida, India

<sup>2</sup> School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India

the average queue size within the target range of the average queue size. The reason is that at a lightly congested link the average queue size varies around minimum threshold ( $min_{th}$ ) when maximum probability ( $max_p$ ) is high while at a heavily congested link the average queue size varies around maximum threshold ( $max_{th}$ ). The Adaptive RED addresses a few issue of RED by adapting  $max_p$  to keep the average queue size within the target range.

The limitations with queue based AQM schemes such as RED, Adaptive RED, and TRED is the direct coupling between queue length and the probability of dropped packet. These AQM schemes are dependent on the load level in steady state queue length. In the case of overloaded flows these schemes suffer not only from higher delay but also from more number of dropped packets [3]. Another controller based scheme i.e. Proportional Integrator (PI) has the ability to decouple both average queue size and the dropping probability [3]. A stochastic optimization based AQM scheme has been presented by Bhatnagar et al. [8].

An adaptive localized active route maintenance (ALARM) mechanism is developed to monitor and maintain the on-going communications by utilizing special control messages techniques which enable fast response to link failure for an existing ad-hoc multipath routing protocol [9]. A novel scheduling approach for multiband mobile routers that can transmit through different frequency bands [10]. The queuing analysis based analytical model is developed by these authors to analyze scheduling algorithm for multiband mobile routers [10]. Learning Automata-Based Congestion Avoidance Algorithm in Sensor Networks (LACAS) addresses the congestion problem in healthcare WSNs [11]. LACAS intelligently learns from the past and it leads to improved performance. Learning-automata-like RED (LALRED) does simultaneously increase the likelihood of the scheme converging to the action which results in reduction of packet drops at the gateway [12]. The LALRED approach improves the performance of congestion avoidance by adaptively minimizing the average queue size and the queue-loss rate. For flow-aware networks (FAN), an efficient congestion control mechanism is developed to enhance the transmission and minimize the delay [13]. An optimized second-order model using robust proportional controller for AQM has been developed in [14].

AQM has been an area of extensive research. Several algorithms have been proposed to handle the congestion at the routers. Recently Karmeshu, Patel and Bhatnagar have proposed an adaptive queue management with random dropping (AQMRD) algorithm which incorporates information not just about the average queue size but also about the rate of change of the same. AQMRD introduces an adaptively changing threshold level that falls in between lower and upper thresholds. This algorithm demonstrates that the additional features significantly improve the system performance in terms of throughput, average queue size, utilization and queuing delay [15].

The queue based AQMs such as RED, MRED, Adaptive-RED, TRED etc. use the average queue size to compute the dropping probability which is modified according to the number of packets to have arrived since the last marked packet. However, it may be pointed out that these algorithms are not efficient in achieving good throughput for congested networks where a sufficient bandwidth is not available. Our aim is to further explore appropriate modification of dropping function to address these limitations.

Our objective in this paper is to find a suitable dropping probability distribution so that the pattern of dropped packets is neither too frequent nor too rare. Based on simulation results, the performance of the proposed scheme is investigated with respect to RED, TRED, Adaptive-RED, and PI. The major contribution of this paper is to address the problem of heavily congested networks. The proposed modified dropping function is able to reduce the delay and

enhance the throughput. The improvement in delay is one of the important requirements of the network. The proposed scheme addresses these issues.

This paper consists of four sections. Section 2 discusses the proposed model and its analysis. The evaluation of the proposed model is carried out in Sect. 3. The simulation results are presented in Sect. 4. Using ns-2 simulator, the performance of the proposed scheme is compared with the existing schemes using the ns-2 simulator. In the last section, we provide conclusion.

## 2 Model Description

### 2.1 Background

The dropping probability is calculated in two steps. In the first step, the dropping probability is calculated as a linear function of the average queue size. In the second step the dropping probability is modified. The packet is dropped in queue size based AQM router where dropping of packet depends on the position of  $q_{avg}$ . If  $q_{avg}$  exceeds the  $max_{th}$  then definitely router drops a packet and if it lies between  $(min_{th}, max_{th})$  then dropping or marking of packet is decided by dropping probability computed at router.

We consider that packets arrive randomly that may be any distribution and some packets are dropped in the manner as shown in Fig. 1. A random variable  $X$  corresponds to the number of packets that arrive after a marked packet until the next packet is marked [1]. The number of unmarked packets is represented by a *count* variable. The random variable  $X$  is assumed to follow different distributions.

#### 2.1.1 Geometric Random Variable $X$

Let  $X$  be a geometric random variable with each packet being marked with probability  $p$ . Then probability that  $k$  packets arrive after a marked packet is

$$P[X = k] = (1 - p)^{k-1}p, \quad k = 1, 2, 3 \dots \tag{1}$$

The corresponding mean and variance are:

$$E[X] = 1/p \text{ and } Var[X] = (1 - p)/p^2 \tag{2}$$

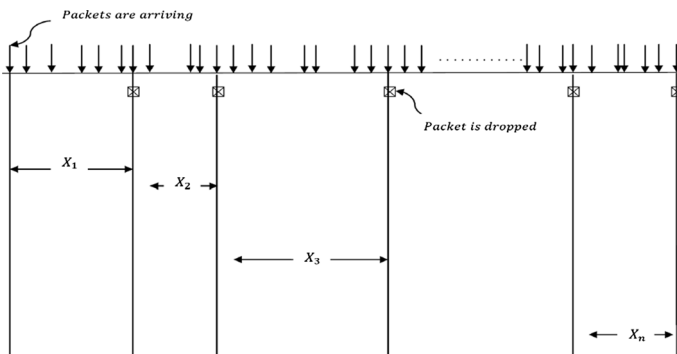


Fig. 1 Model description

### 2.1.2 Uniform Random Variable $X$

In order to render the random variable  $X$  to be uniformly distributed, each arriving packet must be marked with a modified probability  $p/(1 - \text{count} \cdot p)$ . This is an existing model used in RED [1]. The probability that  $k$  number of packets that arrive, after a marked packet, until the next packet marked is given by:

$$P[X = k] = \frac{p}{1 - (k-1)p} \prod_{j=0}^{k-2} \left(1 - \frac{p}{1 - j \cdot p}\right), \quad 1 \leq k \leq 1/p,$$

$$= p,$$

and

$$P[X = k] = 0, \quad k > 1/p$$

The mean number of unmarked packets between two marked packets is

$$E[X] = \frac{1}{2} \left(1 + \frac{1}{p}\right). \quad (3)$$

For detail one can refer [1]. The next subsection present a new dropping function model for computation of dropping probability.

## 2.2 Proposed Model

Let  $X$  be a random variable which represents the number of packets to have arrived after a marked packet until the next marked packet. It also includes a last marked packet. In other way, the variable *count* denotes the number of unmarked packets between two successive marked packets. Therefore *count* will be always  $k - 1$ , if next marked packet is  $k$ th packet. The *count* is set to zero after every marked packet.

Our aim is to find such a marking probability in terms of  $p_1$  and *count* that improves the QoS parameters like throughput, utilization and delay where  $p_1$  is the dropping probability that is calculated using Eq. (5). We set the dropping probability  $p_2$  of each packet as a function of *count* such that  $p_2$  increases when *count* increases. It would be more realistic when the probability of marking increases as the count increases. The modified dropping probability  $p_2$  is calculated according to

$$p_2 = 1 - \frac{p_1(-\log(p_1))}{\text{count} + 1}, \quad \min_{th} \leq q_{avg} \leq \max_{th} \quad (4)$$

where  $p_1$  is calculated as

$$p_1 = \frac{q_{avg} - \min_{th}}{\max_{th} - \min_{th}} \max_p. \quad (5)$$

Here  $q_{avg}$  denotes the average queue length of the packets stored in the router.

Let  $f(p_1) = p_1(-\log(p_1))$ , it can be easily shown that  $f(p_1)$  is always positive and has maxima at  $p_1 = 1/e$  with maximum value  $1/e$ . It can be easily shown that  $\frac{p_1(-\log(p_1))}{j+1} \forall j > 0$  lies between  $(0, 1/e)$  which makes  $p_2$  positive. If  $k$ th packet is marked then *count* reaches to

$k - 1$  since it excludes the last marked packet. Our interest is to identify the probability mass function when the probability of marking is count dependent. The probability that  $k$  number of packets have arrived between two successive marked packets including the last marked packet is given by,

$$P[X = k] = \left( 1 - \frac{p_1(-\log(p_1))}{k} \right) \prod_{j=0}^{k-2} \left( 1 - \left( 1 - \frac{p_1(-\log(p_1))}{j+1} \right) \right), \quad k \geq 1. \tag{6}$$

Let  $X = i$  indicate that  $i$ th packet arrives after a marked packet and  $k$  indicates that  $k$ th packet is marked. Therefore  $i = j + 1$  and  $k$  varies between  $i = 1$  to  $i = k$  where  $j$  varies  $j = 0$  to  $k - 1$ . We can also say that  $k$  includes the marked packet and *count* (represented by  $j$ ) does not include the marked packet. Putting  $a = p_1(-\log(p_1))$ , we have from Eq. (6):

$$P[X = k] = \left( 1 - \frac{a}{k} \right) \prod_{i=1}^{k-1} \frac{a}{i},$$

where  $p_1(i) = a/i$ . After simplification, we get

$$P[X = k] = \frac{a^{k-1}}{(k-1)!} - \frac{a^k}{k!}, \quad k = 1, 2, 3, \dots \tag{7}$$

It would be more realistic to restrict the random variable to assume the maximum value  $C$ . Thus

$$P[X = k] = A \left[ \frac{a^{k-1}}{(k-1)!} - \frac{a^k}{k!} \right], \quad k = 1, 2, 3, \dots, C \tag{8}$$

where

$$A = \left[ 1 - \frac{a^C}{C!} \right]^{-1}, \quad k = 1, 2, 3, \dots, C. \tag{9}$$

The normalization constant  $A \rightarrow 1$  as  $C \rightarrow \infty$  and one can obtains pmf as in Eq. (7). Therefore, we find the expected value of  $X$  as

$$\begin{aligned} E[X] &= A \sum_{k=1}^{\infty} k P[X = k] \\ &= A \sum_{k=1}^{\infty} k \left\{ \frac{a^{k-1}}{(k-1)!} - \frac{a^k}{k!} \right\}, \quad k = 1, 2, 3, \dots, \infty \\ &= A \sum_{k=1}^{\infty} (k-1+1) \frac{a^{k-1}}{(k-1)!} - A \sum_{k=1}^{\infty} k \frac{a^k}{k!} \\ &= A(ae^a + e^a - ae^a) \\ &= Ae^a. \end{aligned}$$

The moment generating function is given as

$$\begin{aligned}
 M_X(t) &= E[e^{tX}] \\
 &= \sum_{k=1}^{\infty} e^{tk} P[X = k] \\
 &= \sum_{k=0}^{\infty} e^{tk} \left( \frac{a^{k-1}}{(k-1)!} - \frac{a^k}{k!} \right) A \\
 &= A(e^t - 1)e^{ae^t},
 \end{aligned}$$

where  $t$  is a parameter. Noting that  $E[X^2] = M_X''(t = 0) = 2Aae^a$  therefore variance can be determined as

$$\begin{aligned}
 \text{Var}[X] &= E[X^2] - (E[X])^2 \\
 &= A(2ae^a - e^{2a}).
 \end{aligned}$$

Therefore,

$$E[X] = Ae^a \quad \text{and} \quad \text{Var}[X] = A(2ae^a - e^{2a}) \quad (10)$$

In terms of  $p_1$  and  $C$ , the mean and variance of random variable  $X$  turns to be:

$$E[X] = \frac{e^{-p_1(\log(p_1))}}{1 - \frac{a^C}{C!}}, \quad (11)$$

$$\text{Var}[X] = -\frac{2p_1 \log(p_1)}{1 - \frac{a^C}{C!}} e^{-p_1(\log(p_1))} - \frac{e^{-2p_1(\log(p_1))}}{1 - \frac{a^C}{C!}} \quad (12)$$

The analysis for mean and variance of dropped packet may be carried out for those packets that accounts for  $q_{avg}$  lies between  $(min_{th}, max_{th})$  using Eqs. (11) and (12).

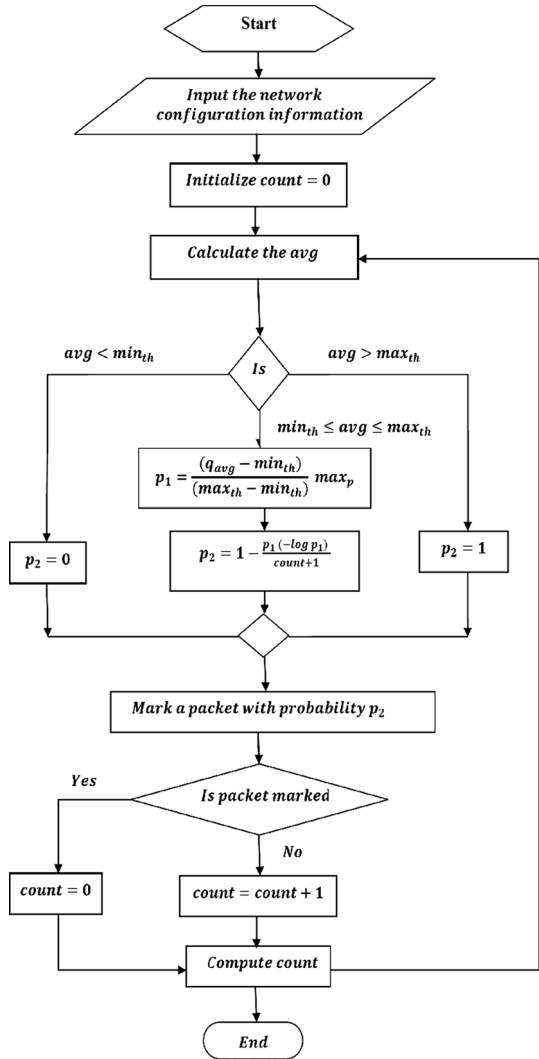
### 2.2.1 Proposed Algorithm

The flowchart for our proposed algorithm for which we implement the code in *ns-2* is shown in Fig. 2. The dropping probability  $p_1$  is calculated in the first step and is computed when the average queue size lies between  $min_{th}$  and  $max_{th}$ . Modified dropping probability calculated from second step is denoted by  $p_2$  and *count* set to zero whenever a packet is marked or dropped. The variable *count* is updated only when the average queue size lies between  $min_{th}$  and  $max_{th}$ . In the following section we investigate the proposed model in relation to the existing models.

### 2.3 Evaluation of Model

We conduct the three experiments when a random variable  $X$  follows: (1) Geometric distribution (2) Uniform distribution (3) Proposed distribution (Eq. (7)). We analyze the models for three choices of random variable  $X$ . The purpose of analysis is to find the

Fig. 2 Flow chart for our proposed scheme



dropping probability distribution which has a smooth dropping pattern of the packets and compare the three mechanisms of marking the packets. Mechanism-1 uses a Geometric random variable, where each packet is marked with probability  $p$ . Mechanism-2 describes a Uniform random variable where each packet is marked with probability  $\frac{p}{1-count \cdot p}$ .

In our proposed mechanism each packet is marked with probability  $1 - \frac{p(-\log p)}{count+1}$ . The choice of parameters is dictated by the fact that the average number of dropped packets remains same for each mechanism. Actually we choose for  $p = 0.02$  (mechanism-1),  $p = 0.01$  (mechanism-2) and  $p = 0.001$  (proposed mechanism). In mechanism-1, mechanism-2, and proposed mechanism about 4489, 4723, 5050 packets are generated and 100 of these packets are marked in each case. In addition to this we also assume that average queue size always lies between  $min_{th}$  and  $max_{th}$  at arrival of each packet. We find that our proposed model achieves smooth dropping of packets at regular intervals as

shown in Figs. 3 and 4 for the three different mechanisms viz., Geometric, Uniform, and the proposed. The Y-axis of Figs. 3 and 4 represents the dropping probability where it has been shown that packet is dropped if dropping probability reaches to 1.

### 3 Simulations and Results

#### 3.1 Experimental Setup

We consider the network topology as shown in Fig. 5. We have proposed a new AQM scheme which has been analyzed for 2D router. The number of input flow entered on 2D plane that can only be accepted. The model can be extended to make useful for 3D router as well even the analysis has been discussed only for 2D router. The gateway/router parameters used in the experiments are set as  $w_q = 0.002$ ,  $max_p = 0.1$ ,  $min_{th} = 20$  packets, and  $max_{th} = 60$  packets. The queue parameter  $min_{th}$  is set at one-third of  $max_{th}$  in each simulation to achieve good performance [5]. A gateway has a buffer size of 64 packets and has a congestion window size of 50 packets. Each packet has a maximum length of 1000-bytes. We perform three experiments keeping the same network parameters but with traffic load changes for three levels (1)  $N = 50$ , (2)  $N = 100$ , and (3)  $N = 200$ . Performance evaluation for AQM schemes is depicted in the following sub-section.

#### 3.2 Performance Evaluation

We evaluate the performance of each AQM scheme for three experiments (1)  $N = 50$ , (2)  $N = 100$ , and (3)  $N = 200$ . Throughput, loss-rate, queue size, and end-to-end delay analysis are calculated.

##### 3.2.1 Throughput Analysis

We perform the simulations for three different network scenarios with network topology shown in Fig. 5. For low traffic and high traffic loads, simulation experiments have been performed and depicted in Figs. 6, 7 and 8. These results show that proposed scheme has better tuning of dropping probability with traffic loads. Most of the packets are marked when the average queue size lies between  $min_{th}$  and  $max_{th}$ . This avoids the situation of exceeding the  $max_{th}$ . The rate of packet drops is more when the average queue size exceeds the  $max_{th}$  which leads to reduction in TCP sending rate and consequently reduces the throughput. We find that AQMRD and Adaptive RED achieve approximately same throughput. The low throughput for RED and TRED is achieved due to very high congested link whereas our proposed scheme achieved high throughput with the aim to control the highly congested networks.

##### 3.2.2 Queue Size Analysis

We perform the experiments for three different traffic loads (1)  $N = 50$ , (2)  $N = 100$ , and (3)  $N = 200$ . A queue size comparison of our scheme with Adaptive-RED for  $N = 50$  is shown in Fig. 9. We find that our scheme is able to stabilize the average queue



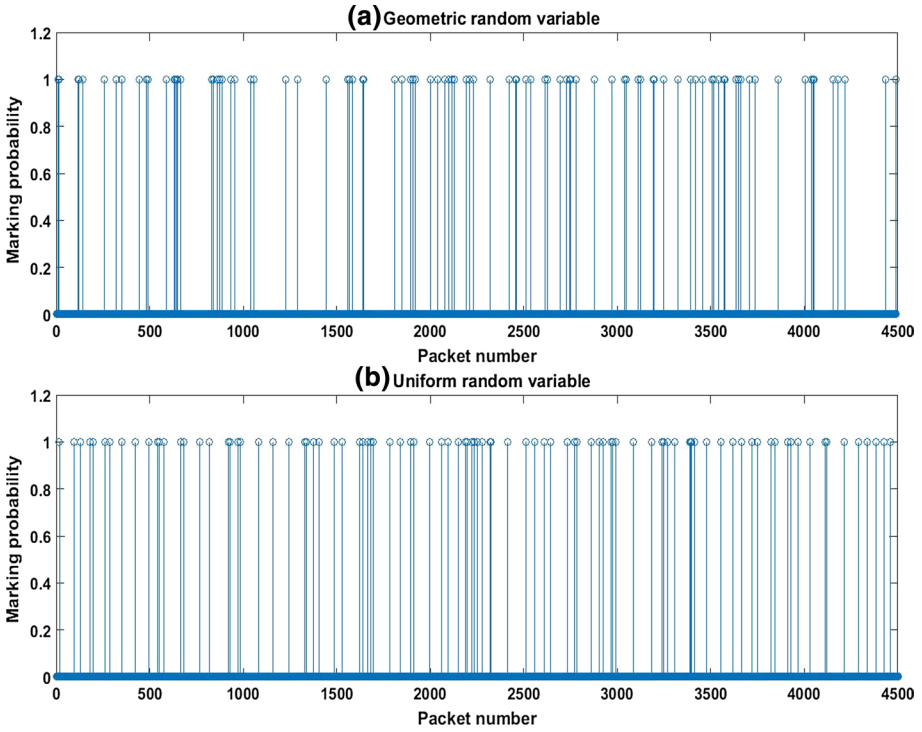


Fig. 3 Dropping pattern for random variable. **a** Geometric, **b** uniform

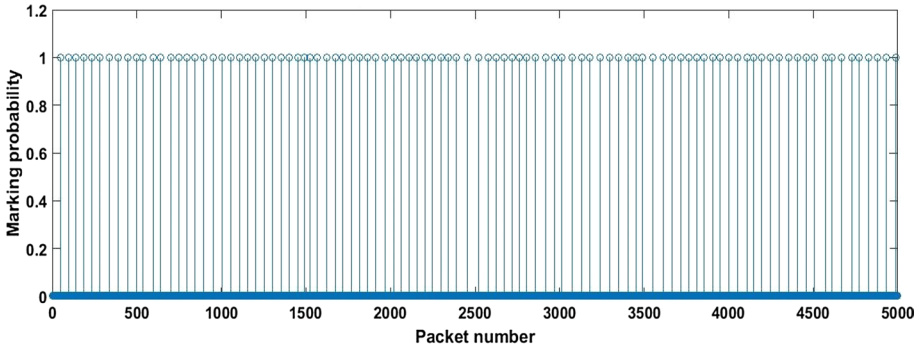


Fig. 4 Dropping pattern for Proposed scheme

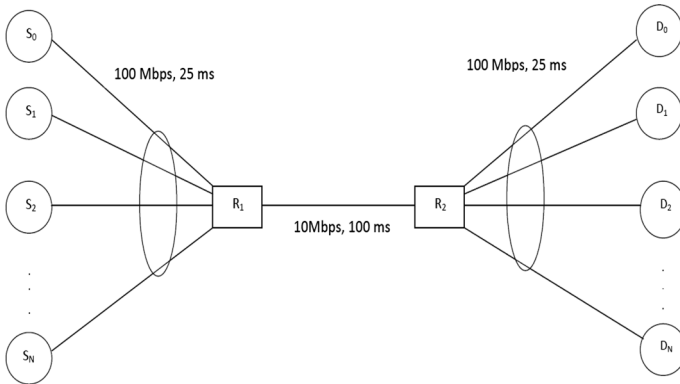


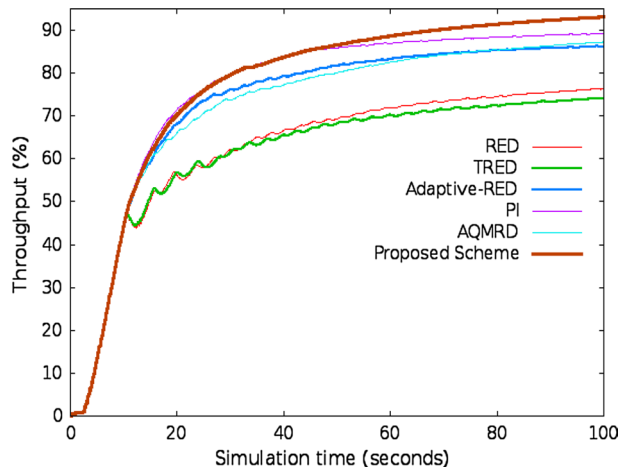
Fig. 5 Dumbbell topology configuration-I

size which suggests a lower end-to-end delay. We compare the current queue size with the PI scheme because PI is the only competing scheme to achieve higher throughput. Figure 10 shows the current queue size comparison with PI, the proposed scheme achieves lower current queue length and the average queue length due to better selection of dropping probability particularly for heavy traffic load.

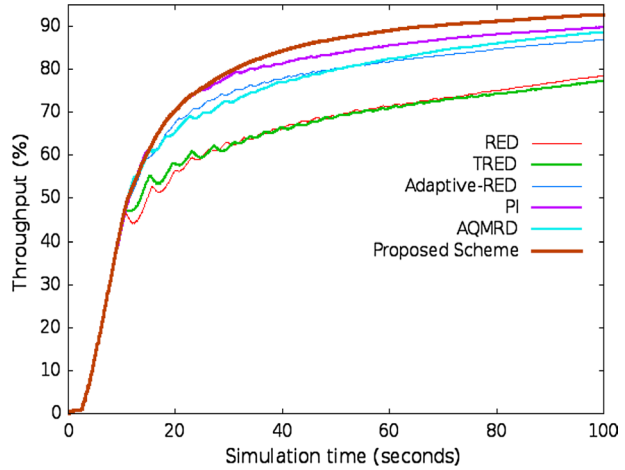
### 3.2.3 End-to-End Delay Analysis

Queuing delay plays an important role under high traffic condition. We have performed simulation experiments to compare the end-to-end delay for each traffic load (1)  $N = 50$ , (2)  $N = 100$ , and (3)  $N = 200$ . Figure 11 shows that our scheme has lowest end-to-end delay in comparison with all other AQM schemes. At other traffic loads  $N = 100$  and  $N = 200$ , our proposed scheme is relatively capable to achieve sufficient reduction in end-to-end delay as given in Figs. 12 and 13. The reduction in delay is achieved due to lesser number of packets stored in the buffer. Thus proposed scheme achieves reduced

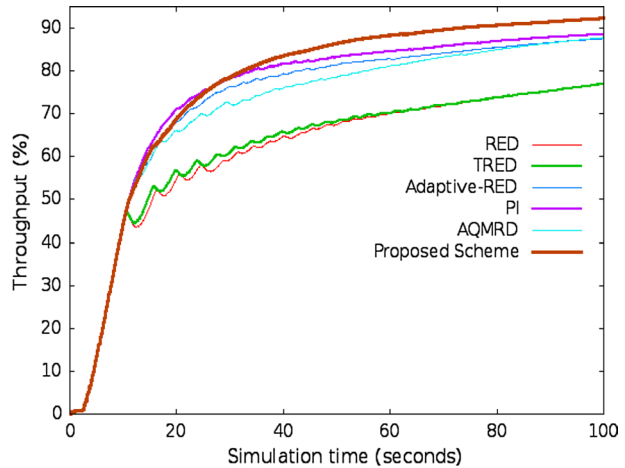
Fig. 6 Throughput comparison for  $N = 50$  (experiment-1)



**Fig. 7** Throughput comparison for  $N = 100$  (experiment-2)



**Fig. 8** Throughput comparison for  $N = 200$  (experiment-3)

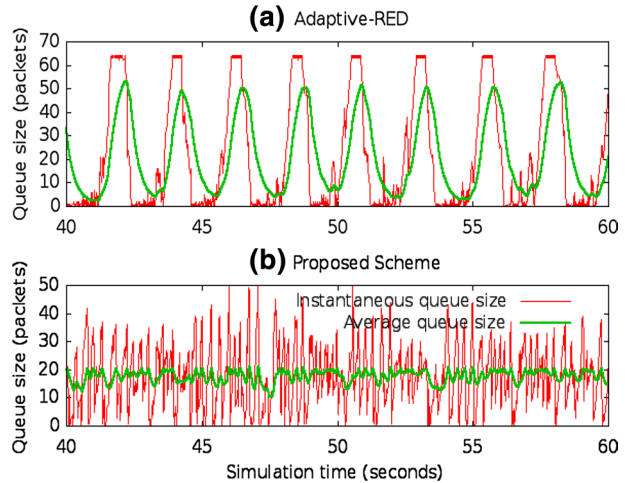


current queue size due to marking of packets with higher probability when packets are arriving after a marked packet.

### 3.2.4 Loss-Rate Analysis

We perform the loss-rate comparison for each AQM scheme. In contrast to two performance metrics viz., throughput and end-to-end delay; our scheme is performing much better in relation to other schemes. However one drawback of our proposed model is that it does not reduce the acceptable loss-rate. The reason is that the our proposed scheme marks a large number of packets to keep lower current queue length for achieving reduced queuing delay. A comparative study of loss-rates is shown in Fig. 14. Similar results are obtained for other traffic loads i.e.  $N = 100$  and  $N = 200$ .

**Fig. 9** Queue size comparison: proposed scheme versus Adaptive-RED (experiment-1)



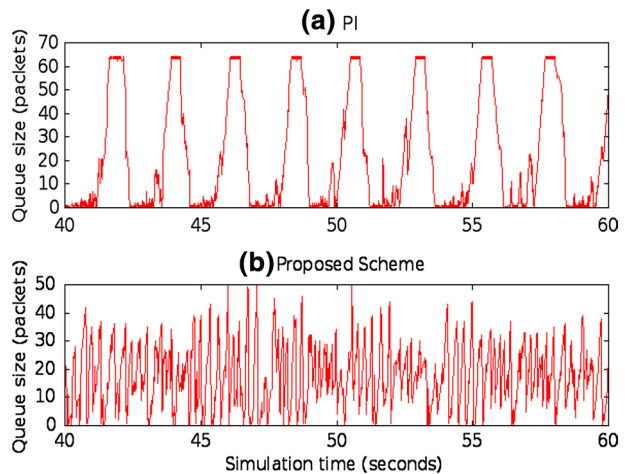
### 3.3 More Experimental Study

In order to test the efficacy of the proposed scheme, we run the simulations for a different network. We perform the experiments for different network topology as shown in Fig. 15. Table 1 shows the network configurations for our experiments. Experiment-4 and experiment-5 are run on network topology-II as shown in Fig. 15. Experiment-4 and experiment-5 have the difference only in link bandwidth, experiment-4 has the same bottleneck link bandwidth while experiment-5 has different bottleneck link bandwidth.

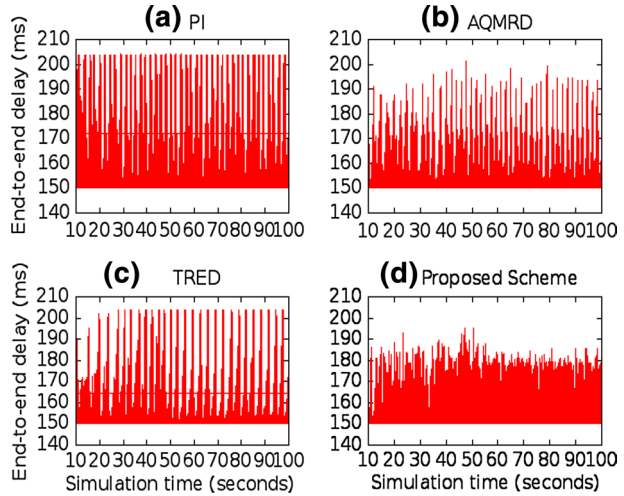
#### 3.3.1 Experiment-4

To check the efficacy of the proposed algorithm we run the simulation for different network topology shown in Fig. 15, with same bottleneck bandwidth between routers  $R_1R_2$

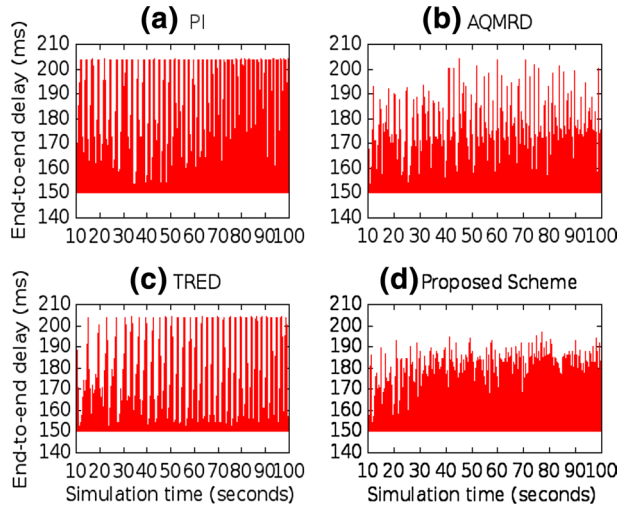
**Fig. 10** Instantaneous queue size comparison: proposed scheme versus PI (experiment-1)



**Fig. 11** End-to-end delay analysis for  $N = 50$  (experiment-1)

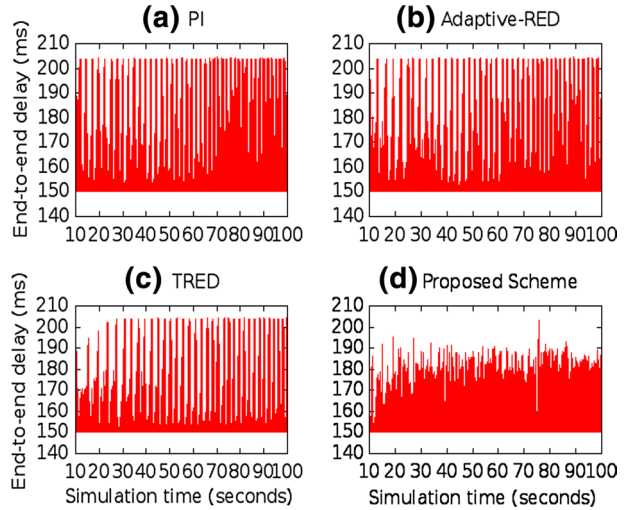


**Fig. 12** End-to-end delay analysis for  $N = 100$  (experiment-2)

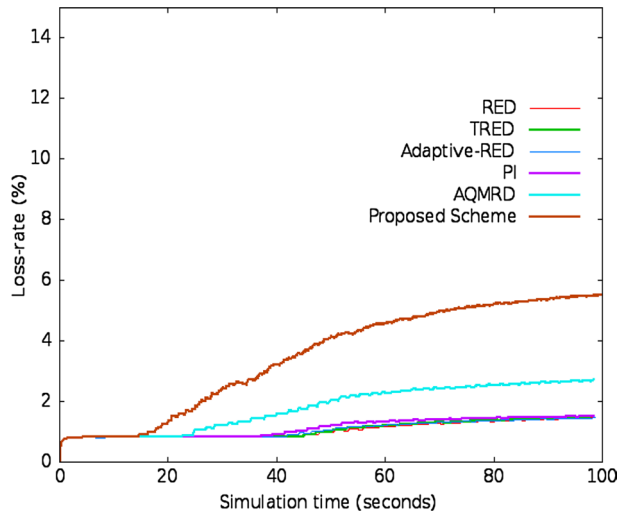


and  $R_2R_3$ . Due to increase in propagation delay between source to destination some of these AQM schemes fail to achieve the higher throughput. Adaptive RED is one of these which shows the drawback of it. Throughput analysis is given in Fig. 16 and we find that the high throughput is achieved by our proposed scheme. This shows that our proposed algorithms is able to handle the congestion irrespective of the changes in propagation delay between the routers or network topology. Due to same flow of data between link  $R_1R_2$  and  $R_2R_3$  having same bandwidth on both links, we do not get any variation in queue size at routers  $R_2$  and  $R_3$ .

**Fig. 13** End-to-end delay analysis for  $N = 200$  (experiment-3)



**Fig. 14** Loss-rate comparison (experiment-1)

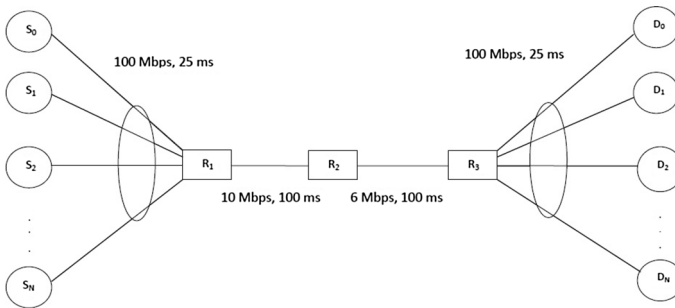


### 3.3.2 Experiment-5

In this experiment, we consider different bandwidths for two bottleneck links of dumb-bell topology shown in Fig. 15. In this experiment we attempt to check the performance of each AQM scheme in heavily congested networks. This network topology generates two level of congestion first at link  $R_1R_2$  and second at link  $R_2R_3$ . Queue size analysis for both current as well as average queue size is compared with Adaptive- RED in Fig. 17. Current queue size comparison with PI is depicted in Fig. 18 and again our scheme yields better stabilization of queue size. We find that throughput of PI, Adaptive RED, and AQMRD suddenly degraded due to very high congestion at link  $R_2R_3$  as shown in Fig. 19. Simulation results show that all the AQM schemes fail to show the robustness with respect to different network topology. Our proposed scheme is more

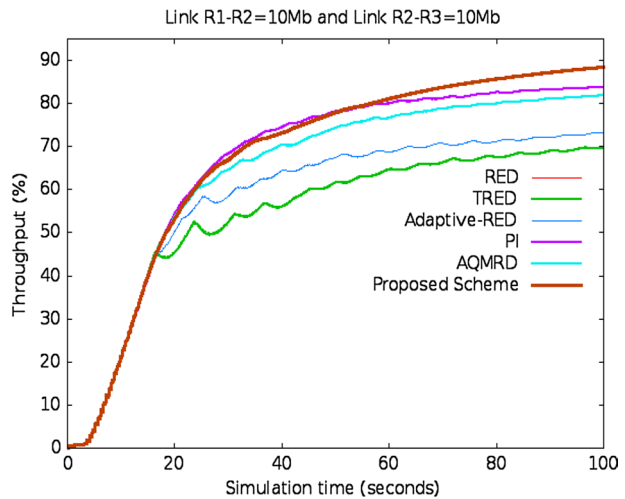
**Table 1** Network configurations

Experiments	Network configurations	Network topology
Experiment-1	$N = 50, B = 64$ , and window size = 50	Network topology-I
Experiment-2	$N = 100, B = 64$ , and window size = 50	Network topology-I
Experiment-3	$N = 200, B = 64$ , and window size = 50	Network topology-I
Experiment-4	$N = 50, B = 64$ , and window size = 50	Network topology-II
Experiment-5	$N = 50, B = 64$ , and window size = 50	Network topology-II



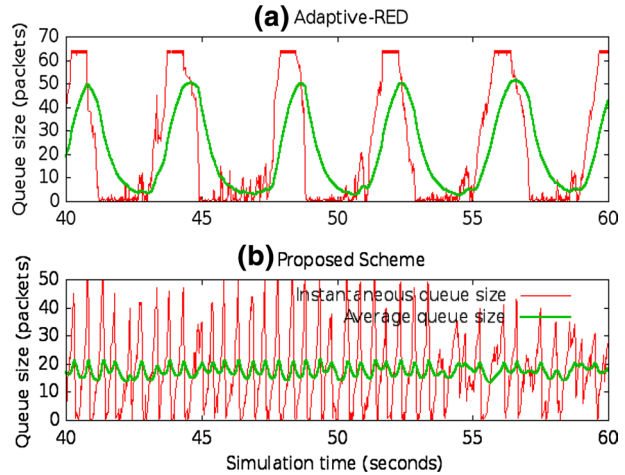
**Fig. 15** Dumbbell topology configuration-II

**Fig. 16** Throughput analysis for experiment-4

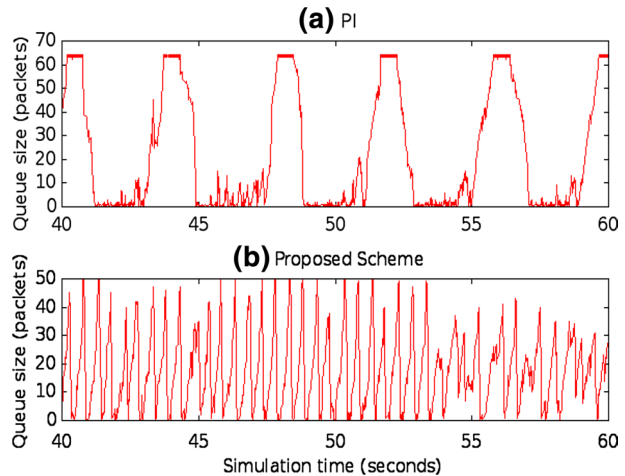


robust than other existing AQM schemes. The achieved throughput is always maintained maximum for our proposed scheme among other AQM schemes irrespective of number of input nodes or sending nodes has been increased. Though the loss-rate increases in such cases when number of input nodes increases. If we use the multiple bottleneck link with different bandwidth then end-to-end delay would be affected. Our scheme shows highest throughput having large difference in relation to other schemes.

**Fig. 17** Queue size comparison: proposed scheme versus Adaptive-RED (experiment-5)



**Fig. 18** Instantaneous queue size comparison: proposed scheme versus PI (experiment-5)

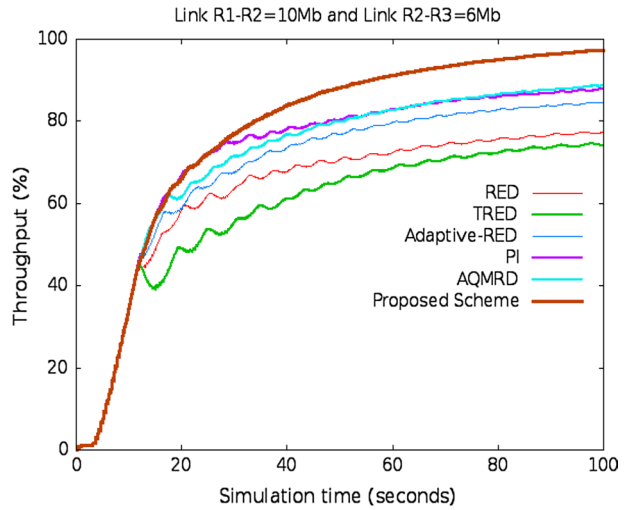


### 3.4 Statistical Analysis of End-to-End Delay

We compare the expected end-to-end delay for each AQM scheme with our proposed scheme. Experiment-1, experiment-2, and experiment-3 are performed on network topology of Fig. 5 while experiment-5 is performed on network topology given in Fig. 15. We find from Table 2 that our proposed scheme achieves lowest mean value of end-to-end delay for every experiment. We observe that the proposed scheme performance increases when load increases because of higher marking probability for those packets which are arriving after a few unmarked packets. The proposed scheme is capable to keep lower current queue length even for heavy traffic load. Thus the objective of proposed scheme, designed particularly for heavy traffic load is achieved.



**Fig. 19** Throughput analysis for experiment-5



**Table 2** Comparative analysis of end-to-end delay

AQM algorithms	End-to-end delay (ms) ( $\mu \pm \sigma^2$ )			
	Experiment-1	Experiment-2	Experiment-3	Experiment-5
Proposed scheme	157.74 $\pm$ 1.70	157.74 $\pm$ 1.89	157.84 $\pm$ 1.85	266.08 $\pm$ 6.33
RED	158.85 $\pm$ 4.48	160.56 $\pm$ 5.13	160.62 $\pm$ 5.22	267.08 $\pm$ 14.69
TRED	159.13 $\pm$ 4.54	160.38 $\pm$ 5.17	160.47 $\pm$ 5.22	268.47 $\pm$ 14.97
Adaptive-RED	160.75 $\pm$ 5.83	162.52 $\pm$ 6.44	162.78 $\pm$ 6.50	270.99 $\pm$ 17.46
AQMRD	156.33 $\pm$ 1.73	157.67 $\pm$ 1.89	157.45 $\pm$ 1.85	262.6 $\pm$ 5.70
PI	162.29 $\pm$ 6.71	164.53 $\pm$ 7.37	163.75 $\pm$ 7.07	274.28 $\pm$ 19.24

### 4 Conclusion

We have performed a number of experiments to check the efficacy of our proposed scheme in relation to others. We use relatively lower bandwidth and higher propagation delay for bottleneck link to create a congested link. We find that our proposed scheme in comparison to RED, TRED, Adaptive-RED, AQMRD, and PI outperforms in terms of throughput. In each experiment, due to high congested link, the performance for RED and TRED is substantially degraded, showing that these schemes are not able to handle the congestion for heavy traffic load.

Our simulation results in Table 2 shows that our scheme yields least end-to-end delay in relation to RED, TRED, Adaptive-RED, and PI except recently proposed AQMRD algorithm [15]. The low variance of end-to-end delay for our proposed scheme shows the stability with respect to delay in the congested networks. The better stabilization of both average queue size as well as current queue size leads to reduction in queuing delay. Due to the consideration of additional parameter i.e. the rate of change of average queue size, AQMRD is able to stabilize the queue size more efficiently.

It is found that PI gives maximum end-to-end delay in each experiment and performs better in terms of throughput but due to highest end-to-end delay it would not be the preferable scheme. The major contribution of this paper is that we have developed a novel AQM algorithm which is able to handle the congestion for highly congested networks. This goal is achieved by our proposed scheme where our concern is more about throughput and delay. In the context performance measures viz., throughput, end-to-end delay, the numerical experiments show that the proposed scheme is found to perform much better than other schemes. The future goal is to design a model for analysis of delay as well as length of the two successive dropped packets.

**Acknowledgements** The author would like to thank Prof. Karmeshu for his useful suggestions and guidance. The help provided by him is highly acknowledged. It will not be possible to complete this paper without his support and time to time discussion with him. We are thankful to the distinguished referee for their comments and suggestions.

## References

1. Floyd, S., & Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4), 397–413.
2. Floyd, S., Gummadi, R., & Shenker, S. (2001). Adaptive RED: An algorithm for increasing the robustness of RED's active queue management, Berkeley, CA. <http://www.icir.org/floyd/papers.html>. Accessed 23 Feb 2017.
3. Hollot, C. V., Misra, V., Towsley, D., & Gong, W. (2001). On designing improved controllers for AQM routers supporting TCP flows. In *Proceedings of IEEE INFOCOM* (pp. 1726–1734).
4. Hollot, C. V., Misra, V., Towsley, D., & Gong, W. (2002). Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Transactions on Automatic Control*, 47(6), 945–959.
5. Feng, G., Agarwal, A. K., Jayaraman, A., & Siew, C. K. (2004). Modified RED gateways under bursty traffic. *IEEE Communications Letters*, 8(5), 323–325.
6. Feng, C. W., Huang, L. F., Xu, C., & Chang, Y. C. (2017). Congestion control scheme performance analysis based on nonlinear RED. *IEEE Systems Journal*, 11(4), 2247–2254.
7. Adams, R. (2013). Active queue management: A survey. *IEEE Communications Surveys and Tutorials*, 15(3), 1425–1476.
8. Bhatnagar, S., Patel, S., & Karmeshu (2018). A stochastic approximation approach to active queue management. *Telecommunication Systems*, 68(1), 89–104.
9. Otero, A. S., & Atiquzzaman, M. (2011). Adaptive localized active route maintenance mechanism to improve performance of VoIP. *Journal of Communications*, 6(1), 68–78.
10. Narman, H., Hossain, M. S., & Atiquzzaman, M. (2015). Management and analysis of multi class traffic in single and multi-band systems. *Wireless Personal Communications*, 83(1), 231–258.
11. Misra, S., Tiwari, V., & Obaidat, M. S. (2009). LACAS: Learning automata-based congestion avoidance scheme for healthcare wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 27(4), 466–479.
12. Misra, S., Oommen, B. J., Yanamandra, S., & Obaidat, M. S. (2010). Random early detection for congestion avoidance in wired networks: The discretized pursuit learning-automata-like solution. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 40(1), 66–76.
13. Domzal, J., Wojcik, R., Cholda, P., Stankiewicz, R., & Jajszczyk, A. (2016). Efficient congestion control mechanism for flow-aware networks. *International Journal of Communication Systems*, 29(4), 787–800.
14. Wang, J., Rong, L., & Liu, Y. (2008). A robust proportional controller for AQM based on optimized second-order system model. *Computer Communications*, 31(10), 2468–2477.
15. Karmeshu, Patel, S., & Bhatnagar, S. (2017). Adaptive mean queue size and its rate of change: Queue management with random dropping. *Telecommunication Systems*, 65(2), 281–295.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Sanjeev Patel** received the B.Tech. degree in Computer Science and Engineering from Motilal Nehru National Institute of Technology, Allahabad, India in 2005; the M.Tech. degree in Computer Science and Technology from Jawaharlal Nehru University, New Delhi, India in 2008; and the Ph.D. degree from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India in 2017. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering at Jaypee Institute of Information Technology, Noida since 2011. Prior to joining this institute, he had worked at Jaypee University of Information Technology, Wahnaghat, Solan, HP, India from 2008 to 2010. He has been served as the Technical Programme Committee in many Conferences. He is an active reviewer for many Journals that includes IEEE Systems Journal, IEEE Transactions on Network and Service Management, International Journal of Communication Systems. He is working in the area of performance modeling of flow and congestion control algorithms, optimization techniques, and machine learning.



**Karmeshu** is a retired Professor of School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India, since 1986. Prior to joining JNU, he taught at the University of Delhi (India), University of Waterloo (Canada), UAM, and UNAM (Mexico). After obtaining his Ph.D. in 1976 in the area of Diffusion of Brownian Particles Through Generalized Langevin Equation, he joined the University of Stuttgart, Germany, as a visiting scientist. In 1993, Dr. Karmeshu received the Shanti Swarup Bhatnagar Award, Indias most prestigious award in the field of science and engineering, for his contributions to mathematical modelling of social and technical systems. He was also awarded the Dr. C. M. Jacob Gold Medal for the year 1990 by the Systems Society of India. Dr. Karmeshu has published about 100 papers in journals. He has been working in the area of modelling and simulation of non-linear stochastic systems covering a wide range of engineering and socio-technical systems. His current research interests lie in the study of mobile and wireless systems, application of entropy framework to communication networks and computational neuroscience.