I J C T A, 10(9), 2017, pp. 765-782 ISSN: 0974-5572

LaQ: A stable loss and queue based active queue management algorithm for congestion control

Sukant Kishoro Bisoy*, Prasant Kumar Pattnaik and Pradeep Kumar Mallick****

ABSTRACT

Active queue management (AQM) technique has been introduced as effective method for congestion avoidance and to achieve better link utilization by reducing the possibility of congestion. In this paper a novel stable AQM technique called LaQ is proposed to control congestion efficiently. During congestion, LaQ adjust the packet drop probability properly to control the router queue. LaQ measures packet loss ratio and uses it with queue length to adjust packet drop probability dynamically. The proposed algorithm achieves stability with faster settling time and robust under dynamic network. The stability of the proposed system is analyzed through control theory technique. The performance of the proposed algorithm is validated by NS2 simulations. Simulation results demonstrate that LaQ is stable with faster responsive and robust to the dynamic network conditions. The proposed algorithm performs better than existing algorithm.

Keywords: AQM; TCP, stability; robustness; settling time; packet loss ratio

1. INTRODUCTION

In congestion control, buffer management for Internet router plays a key role. Droptail is used as a traditional algorithm for buffer management. It drops the incoming packets only when the buffer is full. Due to this passive behavior, the round trip time (RTT) of each packet increases and more importantly correlation between packet drops is lost, resulting in a well know 'TCP synchronization' problem [1]. In recent years, active queue management (AQM) technique has been developed as effective mechanism to mitigate 'TCP synchronization' and to avoid congestion in the network. It plays an important role in improving the performance of TCP in controlling the congestion. The main objective of AQM is to signal congestion early before the buffer is full. Due to this active behavior, AQM is able to reduce congestion and overcome the synchronization problem with the proper selection of packet dropping/marking function. The primary goal of AQM is: low queuing delay, high link utilization with lower packet loss. Other goals of AQM are to make the system stable and robust against different system parameters. Random early detection (RED)[2] is used as widely deployed AQM technique, where queue length is used as a measure of congestion. Due to its popularity, RED is studied well in the literature. Even though it is studied widely in the literature, the behavior is not fully understood and effectiveness of RED relies on parameter setting. Based on RED other mechanisms such as Adaptive RED (ARED) [3], Refined Adaptive RED (Re-ARED)[4], Nonlinear RED (NL-RED)[5], Stabilized RED (SRED)[6], Cautious Adaptive RED (CARED) [7], Loss-ratio based RED (LRED) [8], Exponential RED[9] and Dynamic RED (DRED) [10] have been developed. Although RED can overcome global synchronization and blocking problems, there is a performance difference for different

* Research Scholar, Department of Computer Science and Engineering, Siksha 'O' Anusandhan University, Bhubaneswar 751030, Odisha, India, *Email: sukantabisoyi@yahoo.com*

** School of Computer Engineering, KIIT University, Bhubaneswar, Odisha, India.

*** Department of Computer Science & Engineering, Vignana Bharathi Institute of Technology, Hyderbad, *Email: pradeepmallick84@gmail.com*

values of the parameter. The heuristic approaches such as RED and its variants require proper tuning of its parameters to get better performance. In the recent literature many AQM techniques have been developed [11], [12], [13], [14], [15], [16], [17], [18], [19], [20],[21], [22] and [23]. A detailed survey of AQM controllers is presented in [24]. One of the important functions of AQM is to measure the arrival rate of incoming traffic and control the queue length where the nature of the traffic is bursty. To regulate the queue length efficiently an appropriate packet drop probability function is required. Author in [25] proposed an AQM algorithm to predict change in packet arrival rate and define appropriate packet drop probability function based on that prediction. Recently, much attention has been given to the stability and robustness analysis of AQM under dynamic environments. To ensure stability of queue length an AQM scheme is designed in [26] and control theory is used to overcome the nature of multiple delays. Wang et al. in [27] studied the stability of AIMD/RED system using multi-bottleneck topology and developed a mathematical model to analyze the stability of the system. Recent study has shown that using queue length with input traffic rate can enhance the stability of the system.

In this paper an AQM algorithm called LaQ has been proposed to enhance the stability of the system by controlling both queue length and loss rate at the router. It measures packet loss ratio and uses it with current queue length to define packet drop probability dynamically. It maintains its queue length around target and achieves faster response with lower average delay. Then the performance of LaQ is compared with ARED, REM and PI algorithm in dynamic network environment.

The rest of the paper is organized as follows. Section 2 describes the proposed LaQ algorithm. Section 3 presents fluid flow model of network dynamics and simulation results in section 4. Finally section 5 concludes the paper.

2. THE PROPOSED ALGORITHM

This section presents the design of proposed LaQ algorithm based on packet loss ratio and queue length measured at each sampling time(δ). Aim of LaQ is to adjust packet drop probability to regulate the queue length around the desired operating point. It brings stability in the system by controlling the resources of the router properly. At equilibrium point the packet loss ratio comes close to the packet drop probability. It adopts a dropping policy based on a dynamic model of the network. The dropping policy depends on both loss ratio and queue length measurement. It measure packet loss ratio at each sampling time and use it with queue length to find packet drop probability. At each sampling interval, LaQ find the queue length and estimate packet loss ratio as estimated in [8] to compute packet drop probability (p). At rth measurement, let the packet loss ratio is $l(r)$. Let $p_d(r)$ is the number of packet dropped and $p_a(r)$ is the number of packet arrived. The packet loss ratio is measured using Eq. (1) during latest M period.

$$l(r) = \frac{D}{A}, \tag{1}$$

where $D = \sum_{i=0}^{M} p_d(r-i)$ and $A = \sum_{i=0}^{M} p_a(r-i)$. Based on $l(r)$, the estimated packet loss ratio $\overline{l(r)}$ is calculated using exponential weighted moving average (EWMA) as average queue length is calculated in RED algorithm and shown in Eq. (2).

$$\overline{l(r)} = (1-w_m).l(r) + \overline{l(r-1)}.w_m, \tag{2}$$

where w_m is the weighting factor. In LaQ, a very simple packet drop probability function is used which is based on both estimated packet loss ratio and current queue length. The packet drop probability is defined as:

$$p(r) = \overline{l(r)} + \gamma\left(q(r) - Q_{ref}\right), \tag{3}$$

where Q_{ref} is target queue length. The packet loss ratio and packet drop probability is updated at the same time scale i.e. sampling period (δ). At equilibrium point, LaQ tries to match the queue length around target. The detail of proposed algorithm is given in Figure 1.

/* Parameters */

q: Current queue length

γ: Constant

Loss Ratio: Estimated packet loss ratio

M: Number of latest period to measure *Loss Ratio*

Q_{ref}: Target queue length

p: Packet dropping probability

w_m: Measurement weight

/* Initialization */

index = 0, *arrPktNum* = 0, *dropPktNum* = 0, *LossRatio* = 0.0, w_m = 0.1, M = 10

Measure_LossRatio( ) /*Called inside Calculate_P( ) function */

```
Drop_Num[index] = Drop_Num[index] = dropPktNum;
Arr_Num[index] = arrPktNum;
dropPktNum = 0;
arrPktNum = 0;
index++;
if(index == M) index = 0;
for(j = 0; j <M; j ++) total_Drop+=Drop_Num[j]; }
for(j = 0; j <M; j ++) total_Arr+=Arr_Num[j]; }
LossRatio_Temp = total_Drop/total_Arr;
LossRatio = LossRatio *w_m + LossRatio_Temp (1 – w_m);
total_Drop = total_Arr = 0;
```

Calculate_P( ) /*Called every δ seconds */

```
Measure_LossRatio( );
q;  /* measure current queue length */
p = LossRatio + γ(q – Q_ref);
if (p > 1) p = 1;
if (p < 0) p = 0;
```

Enqueue( ) /* Called upon each packet arrival */

```
arrPktNum + +;
u = uniformRand (0, 1);
if ( buffer is full) {
Drop the packet;
 dropPktNum++;
}
else if (u>p){
Enqueue the packet;
arrPktNum++;
}
else {
Drop the packet;
dropPktNum++;
}
```

Figure 1: The LaQ Algorithm.

3. FLUID MODEL FOR NETWORK DYNAMICS

For network flow dynamics, several theoretical models have been proposed in the literature. The TCP behavior was modeled in [28] using stochastic differential equation analysis. According to this model a dumbbell network topology is created where all the sources use TCP as their transport layer protocol as shown in Figure 2. For simplicity, we assume the system is homogeneous where $W(t)$ is the TCP window size and $q(t)$ is the queue length in packets and $p(t)$ is the packet drop probability. Let $R(t)$ be the round-trip time of each TCP connection in second and C is the link capacity of single bottleneck link. Let T_p be the propagation delay, $r(t)$ be the input traffic rate and $N(t)$ be the number of TCP connections.

A network can be modeled as "feedback control system" using control theory. In control theory, endpoint (Ex. TCP Process) along with a router can form "plat" which is to be controlled by any controller. AQM algorithm can be used as controller, which provides packet drop probability as a control signal to the plant to adjust the source route. The AQM controller can be designed using control theory as shown in Figure 3. The queue length can be used as controlled variable and Q_{ref} can be used as target queue length (reference input) to the system. Based on current queue length and reference input, the error signal $e(t)$ is calculated to help the AQM controller in controlling the plant. At steady state the error signal is zero.

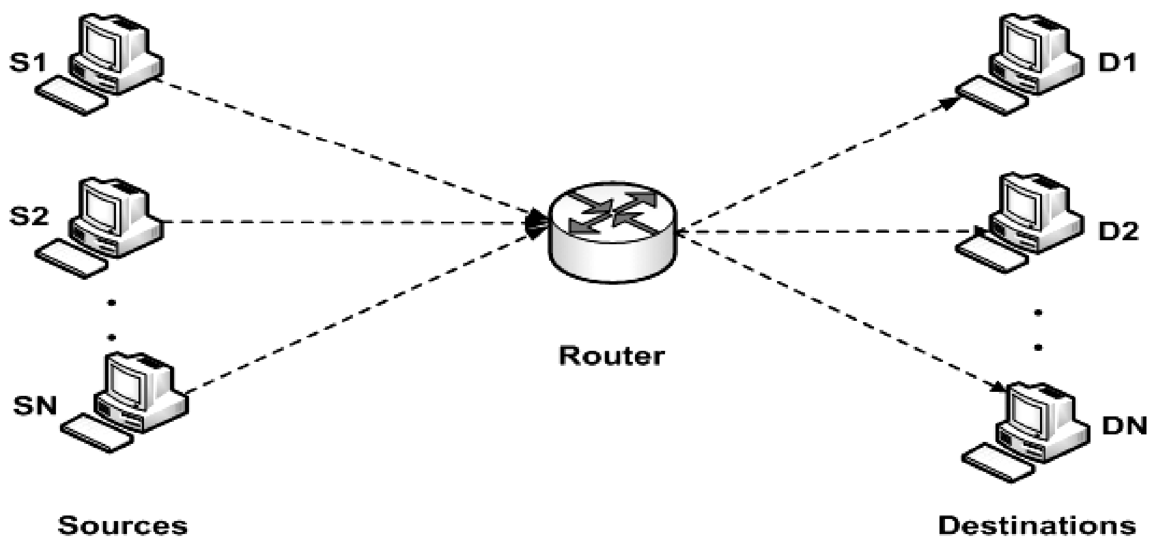


Figure 2: Network Model

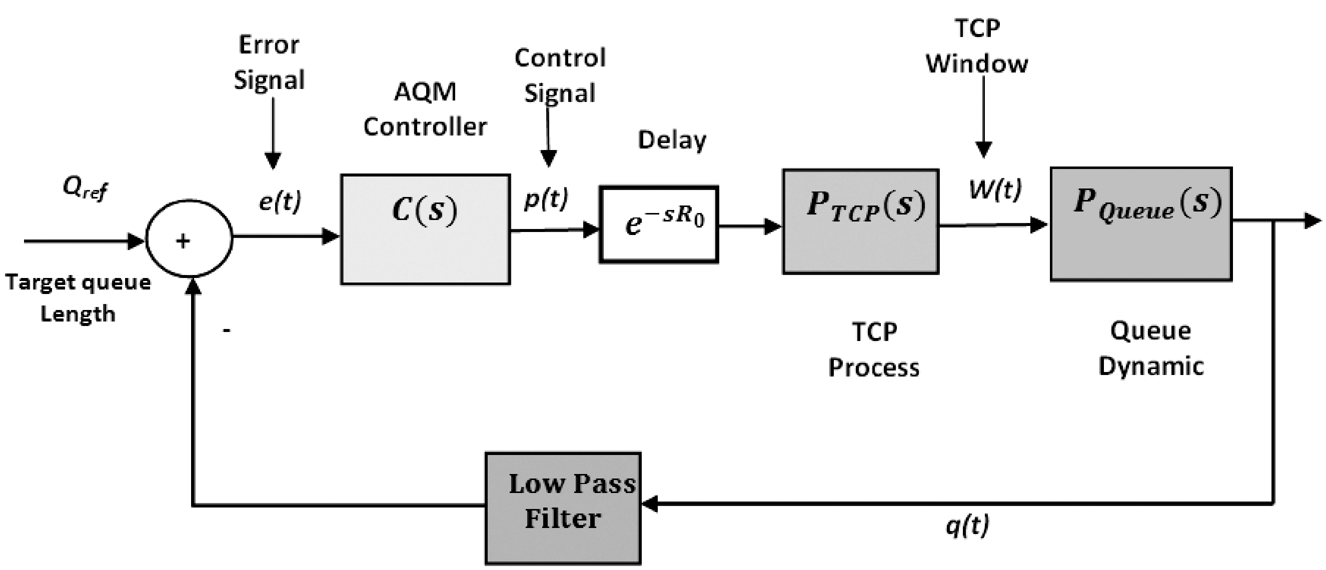


Figure 3: AQM design using control theory concepts

We followed the model presented in [28]. The model is described by Eq.(4) and (5).

$$\frac{dW(t)}{dt} = f(W,q,p) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} p(t-R(t)) \tag{4}$$

$$\frac{dq(t)}{dt} = g(W,q,p) = N(t)\frac{W(t)}{R(t)} - C \tag{5}$$

On the other hand, the packet drop probability of LaQ is defined as

$$\frac{dp(t)}{dt} = h(W,q,p) = p(t) + \gamma\left(q(t) - Q_{ref}\right) \tag{6}$$

The queue length q and window size W are positive and bounded quantities i.e. $q \in [0,\bar{q}]$ and $W \in [0,\bar{W}]$ where $\bar{q}$ and $\bar{W}$ denote buffer capacity and maximum window size respectively.

Assuming $N(t) \equiv N$ and $R(t) \equiv R_0$ as constants taking (W, q) as state as input, the operating point (W_0, q_0, p_0) is defined by

$$\frac{dW(t)}{dt} = f(W,q,p) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} p(t-R(t)) = 0 \tag{7}$$

$$\Rightarrow \frac{W(t)W(t)}{2} p_0 = 1$$

$$\Rightarrow W_0^2 p_0 = 2$$

and $q(t) = 0 \Rightarrow N(t)\frac{W(t)}{R(t)} - C = 0$

$$\Rightarrow W_0 = \frac{R_0 C}{N}$$

We linearize Eqs. (4)-(6) about the operating point using Taylor series approximation.

Let $\delta W = W - W_0, \delta q = q - q_0, \delta p = p - p_0$

$$\frac{d\delta W(t)}{dt} = \frac{\partial f}{\partial W}\delta W + \frac{\partial f}{\partial q}\delta q + \frac{\partial f}{\partial p}\delta p \tag{8}$$

$$\frac{d\delta q(t)}{dt} = \frac{\partial g}{\partial W}\delta W + \frac{\partial g}{\partial q}\delta q + \frac{\partial g}{\partial p}\delta p \tag{9}$$

$$\frac{d\delta p(t)}{dt} = \frac{\partial h}{\partial W}\delta W + \frac{\partial h}{\partial q}\delta q + \frac{\partial h}{\partial p}\delta p \tag{10}$$

This implies

$$\frac{d\delta W(t)}{dt} = K_{11}\delta W + K_{12}\delta p(t-R) \quad (11)$$

$$\frac{d\delta q(t)}{dt} = K_{21}\delta W + K_{22}\delta q \quad (12)$$

$$\frac{d\delta p(t)}{dt} = \delta p + K_{31}\delta q \quad (13)$$

Let $K_{11} = \frac{-2N}{R^2C}, K_{12} = \frac{-RC^2}{2N^2}, K_{21} = \frac{N}{R}, K_{22} = -\frac{1}{R}$ and $K_{31} = \gamma$.

The stability of the proposed system is analyzed by obtaining characteristic equation. Performing Laplace transforms of Eqs. (11) (12) and (13), we obtain:

$$sW(s) = K_{11}W(s) + K_{12}P(s)e^{-sR} \quad (14)$$

$$sQ(s) = K_{21}W(s) + \frac{1}{R}Q(s) \quad (15)$$

$$sP(s) = P(s) + K_{31}Q(s) \quad (16)$$

where $W(s)$, $Q(s)$ and $P(s)$ represents the Laplace transform of $W(t)$, $q(t)$ and $p(t)$, respectively.

Next, we determine the characteristic equation $A(s)$ of the network system from the Eqs. (14)-(16). To compute the value of the characteristic equation $A(s)$, the second order approximation $e^{-sR} \approx 1 - sR + s^2R/2$ is used. By using Eqs.(14)-(16) and second order approximation we have the characteristic equation:

$$A(s) = s^3 + s^2\left[\frac{1}{R} + \frac{2N}{R^2C} + \frac{C^2RK_{31}}{4N} - 1\right] + s^1\left[\frac{2N}{R^3C} - \frac{1}{R} - \frac{2N}{R^2C} - \frac{C^2RK_{31}}{2N}\right] + s^0\left[\frac{C^2K_{31}}{2N} - \frac{2N}{R^3C}\right] \quad (17)$$

For the above Eq. (17), we let the coefficients of s^3, s^2, s^1 and s^0 are a_3, a_2, a_1 and a_0, respectively. We have:

$$a_3 = 1 \quad (18)$$

$$a_2 = \frac{1}{R} + \frac{2N}{R^2C} + \frac{C^2RK_{31}}{4N} - 1, \quad (19)$$

$$a_1 = \frac{2N}{R^3C} - \frac{1}{R} - \frac{2N}{R^2C} - \frac{C^2RK_{31}}{2N} \quad (20)$$

$$a_0 = \frac{C^2K_{31}}{2N} - \frac{2N}{R^3C}, \quad (21)$$

Based on Eq. (17) the characteristic equation $A(s)$ can be written as:

$$A(s) = a_3s^3 + a_2s^2 + a_1s^1 + a_0s^0. \quad (22)$$

The system is stable, if all the zeros of $A(s)$ are in open left-half plane (OLHP). For the Eq. (22), the Routh table is constructed using Routh-Hurwitz theory and shown in Table 2.

Table 2
The Routh Table.

s^3	a_3	a_1	0
s^2	a_2	a_0	0
s^1	$\frac{a_2 a_1 - a_3 a_0}{a_2}$	0	0
s^0	a_3	0	0

For simplicity, let

$$L_1 = \frac{a_2 a_1 - a_3 a_0}{a_2}. \tag{23}$$

Routh stability test states that the system is stable if and only if all the values present in the second column of Table 2 are greater than zero(positive), i.e. :

$$a_3 > 0,\ a_2 > 0,\ L_1 > 0 \text{ and } a_0 > 0. \tag{24}$$

Example 1: Consider the case for a network having parameters: $N = 250$, $C = 20Mb = 2500$ packets/sec, $R = 0.08$ sec, $K_{31} = \gamma = 0.05$.

From Eqs.(18)-(21), we have, $a_3 = 1$, $a_2 = 68$, $a_1 = 296.625$ and $a_0 = 234.375$. For the system stability, all the elements present in the second column of Table 3 must be positive. The values of Table 2 are computed using above values of the different network parameters and shown in Table 3.

Table 3
The Routh Table for Example 1.

s^3	1	296.625	0
s^2	68	234.375	0
s^1	293.178	0	0
s^0	234.375	0	0

As the values of second column of Table 3 is greater than zero(Positive), the proposed system is stable.

4. SIMULATION RESULTS

The performances of LaQ algorithm is evaluated and compared with existing AQM algorithms through a number of experiments using NS2 simulator [29]. We use the queue length (packets), average delay (ms), packet loss ratio, average jitter, mean and standard deviation of queue length and settling time (sec) as the performance index. The performance of LaQ is evaluated in various scenarios using single bottleneck and multi bottleneck link topology. Initially a single bottleneck topology is created as shown in Figure 4, where 250 TCP connections (N) pass through the common link between router R1 and router R2. The bandwidth and delay of side links are 100Mb and 10ms, respectively. The bottleneck link capacity (C) is 20Mb (2500 packets/sec) and propagation delay is 20ms. Each TCP connection has equal round trip time (R = 0.08 sec). The mean packet size is 1000 bytes and the buffer size is 600 packets. The router R1 uses LaQ and other router use Droptail to manage the queue. All sources use TCP/Reno as end to end congestion control mechanism and generate greedy file transfer protocol (FTP) traffic at the same time. Each simulation runs for 100 sec. The reference queue length (Q_{ref}) is 200 packets. For ARED, we set $min_{th} = 100$, $max_{th} = 300$, $\alpha = 0.0$, $\beta = 0.9$ and $interval = 0.5$. For PI, we set $\alpha = 0.00001822$, $b = 0.00001816$ and sampling frequency (δ) is 170. For REM, we use default setting.

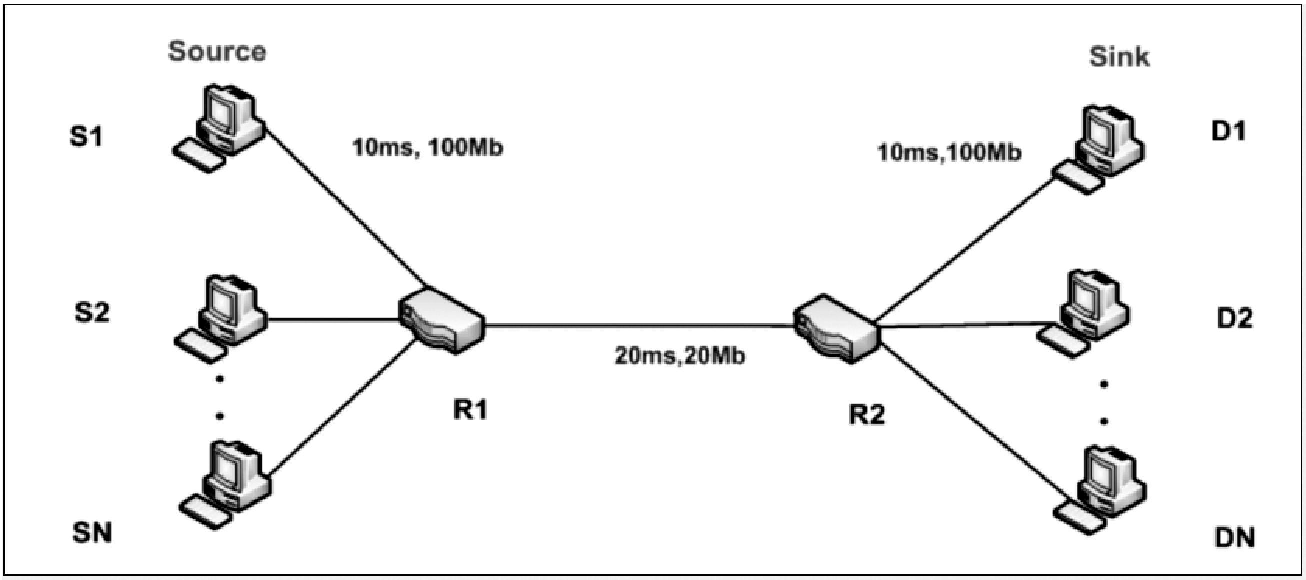


Figure 4: The single bottleneck topology.

For LaQ, we set control parameter (γ) = 0.05, w_m = 0.1, M = 10 and the sampling frequency is 170, to ensure the stability of the system.

We compare the performance of proposed algorithm with existing algorithms ARED, REM and PI under various scenarios.

4.1. Stability under various scenarios

In this section, we compare the performance of LaQ with the existing AQM algorithms, ARED, REM and PI using single bottleneck topology shown in Figure 4.

4.1.1. Experiment 1: Comparison of AQM algorithms

In this experiment we test the performance of LaQ and compare with ARED, REM and PI under homogeneous scenario for 250 TCP connections. The quantitative comparisons of AQM algorithms are summarized in Table 1. As illustrated in the table, LaQ achieve higher throughput, higher link utilization with lower delay than others. The average delay of ARED is lower than others. However it achieve lower throughput, lower link utilization with higher packet loss ratio. There is always trade-off between link utilization and delay. With respect to all parameters we have considered, the

Table 1
Performance of AQM algorithms.

AQM Algorithms	*LaQ*	*ARED*	*REM*	*PI*
Link utilization (%)	96.04	96.00	96.04	96.04
Throughput (Mbps)	19.20952	19.20104	19.20952	19.20952
Average delay (ms)	0.082711	0.061119	0.087979	0.113763
Settling time (sec)	1.99858	–	25.62280	90.11812
Packet loss ratio (%)	16.53	18.04	15.63	12.61
Average jitter (ms)	0.000222	.000166	0.000940	0.000866
Mean queue length (Packets)	203.542	0106.959	229.156	352.148

performance of LaQ is better than others. The mean queue length of proposed algorithm is close to the target with very small settling time than ARED, REM and PI. LaQ require 1.99858 sec to bring the queue length into steady state. It shows that the settling time of proposed algorithm is faster than others. However, ARED is not able to maintain its queue length around target (200 packets). Table 1 illustrates that proposed algorithm LaQ guarantee quality of service (QoS) by achieving higher link utilization and higher throughput and reducing average delay and settling time. The settling time is the amount of time required to move from the oscillatory behavior of queue length to steady state condition. A smaller settling time represents the responsive connection. For the stability of the system, maintaining steady state around target with small settling time is required.

Then the stability of AQM algorithms is analyzed by measuring queue length for the whole simulation duration. The queue length dynamics of each AQM algorithm for 250 TCP connections is shown in Figure 5. We set the target queue length of each AQM at 200 packets. As shown in Figure 5, each algorithm can stabilize the queue length around target with different settling time except ARED. However, LaQ converge the queue length around target with faster response. It is demonstrated that LAQ is more efficient in controlling the queue length around target than others.

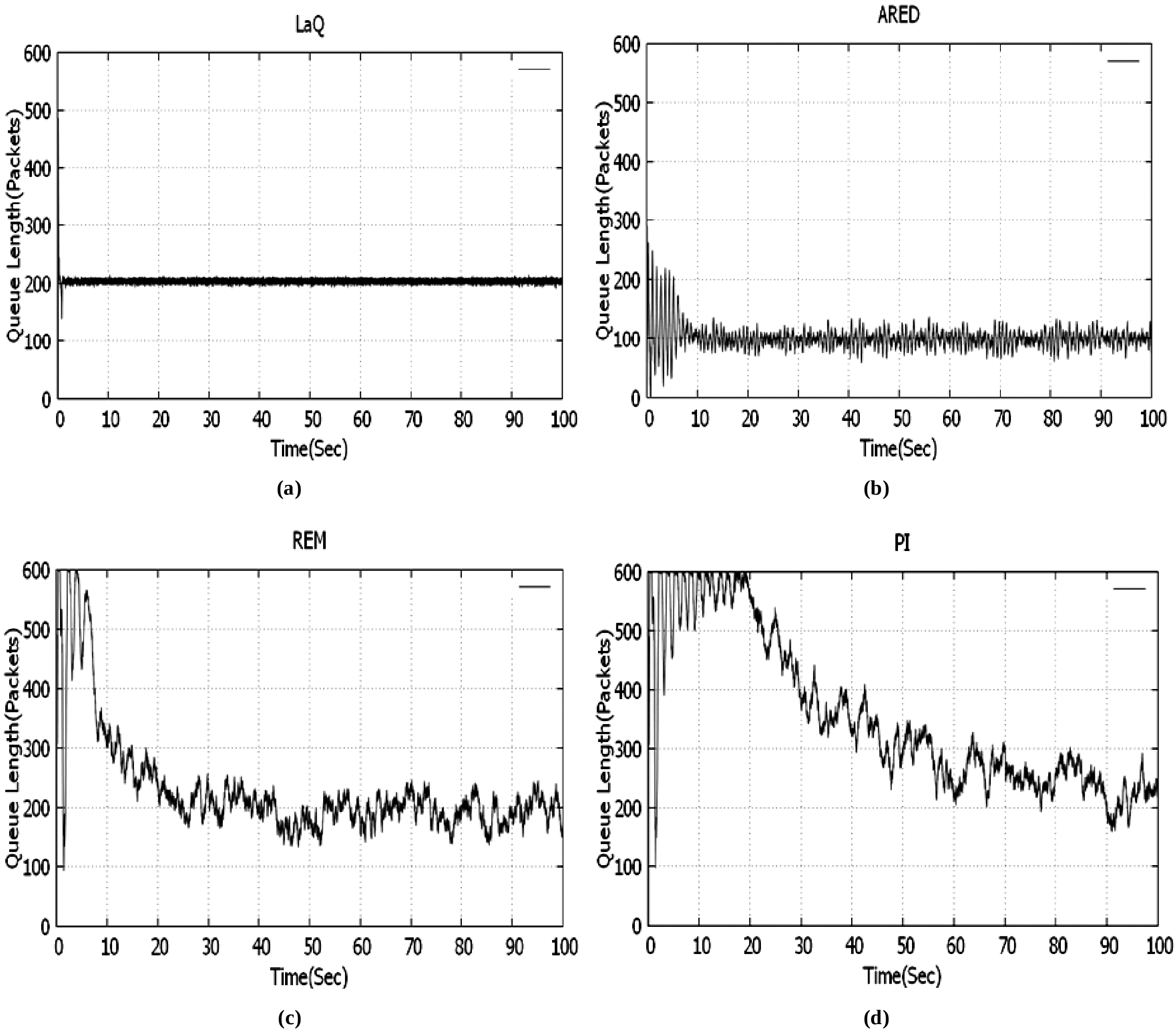


Figure 5: Queue length of (a) LaQ (b) ARED (c) REM (d) PI.

4.1.2. Experiment 2: Stability under sudden change of TCP traffic

In this section we test the stability AQM algorithms under sudden TCP traffic change. During simulation, TCP traffics are added and dropped abruptly in the networks. At beginning, 250 numbers of TCP connections are created and 100 additional TCP connections are arrived to the link at 50 sec. So total number of TCP connections in the networks is 350 at 50 sec. At 70sec, same 100 numbers of TCP connections are dropped from the network. The queue length of AQM algorithms are shown in Figure 6. The result shows that LaQ efficiently stabilizes the queue length around target under time varying TCP traffic. It is less sensitive to the traffic present in the network. The proposed algorithm manages well to keep its queue length around target even when the TCP traffic varies dynamically. Proper adjustment of the packet drop probability of LaQ helps to achieve the stability in queue length. On the other hand, the queue length fluctuations of PI and REM are very large during traffic change as compared to ARED and LaQ. From this experiment, it is observed that the proposed algorithm is stable under sudden change TCP traffic.

4.1.3. Experiment 3: Stability under random start time

In this section we test the stability of AQM considering 250 TCP connections whose starting time is random between 0 to 10 sec. The queue length dynamics of each AQM algorithm are shown in Figure 7. In this the

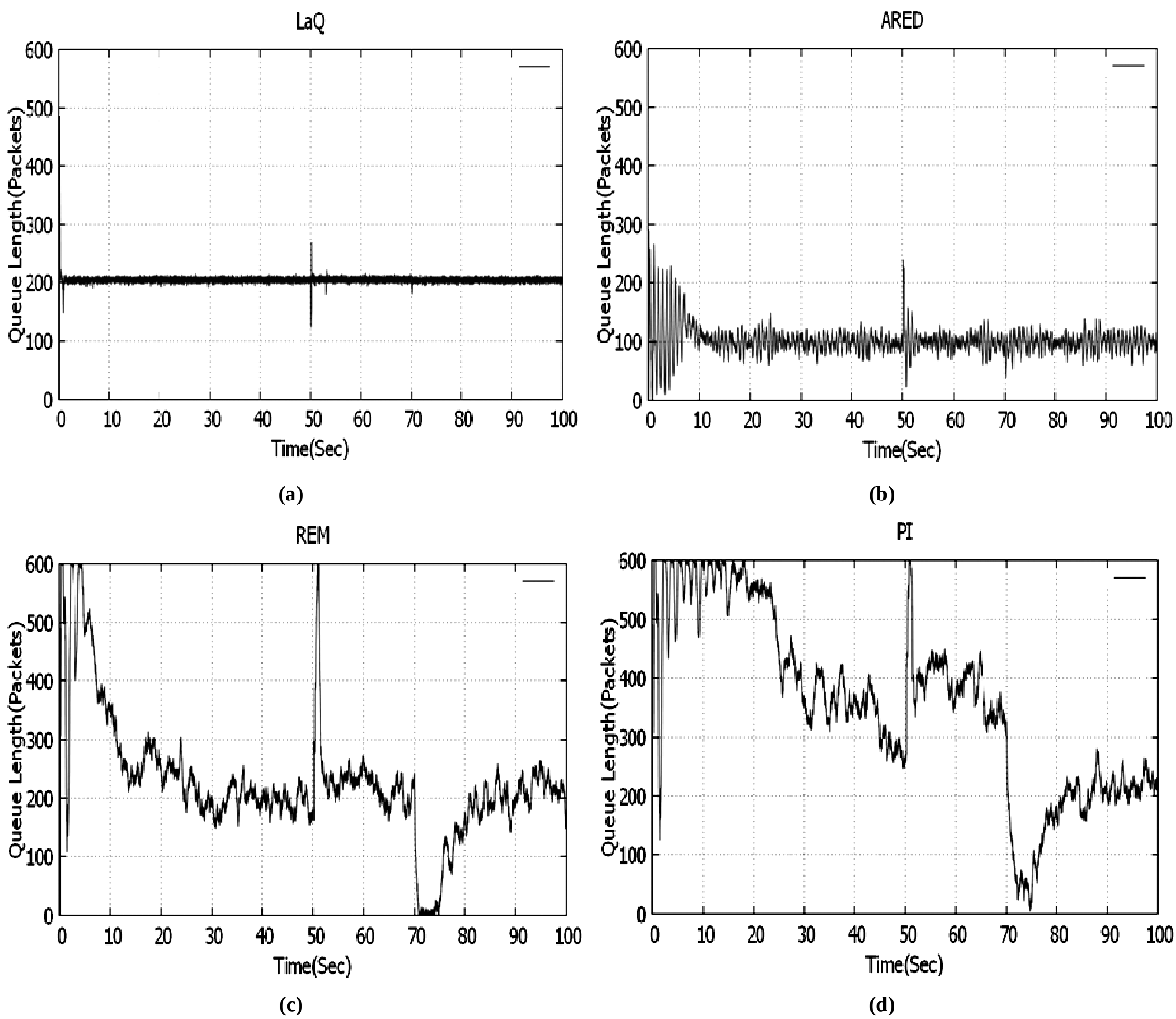


Figure 6: Queue length under sudden traffic change of (a) LaQ (b) ARED (c) REM (d) PI

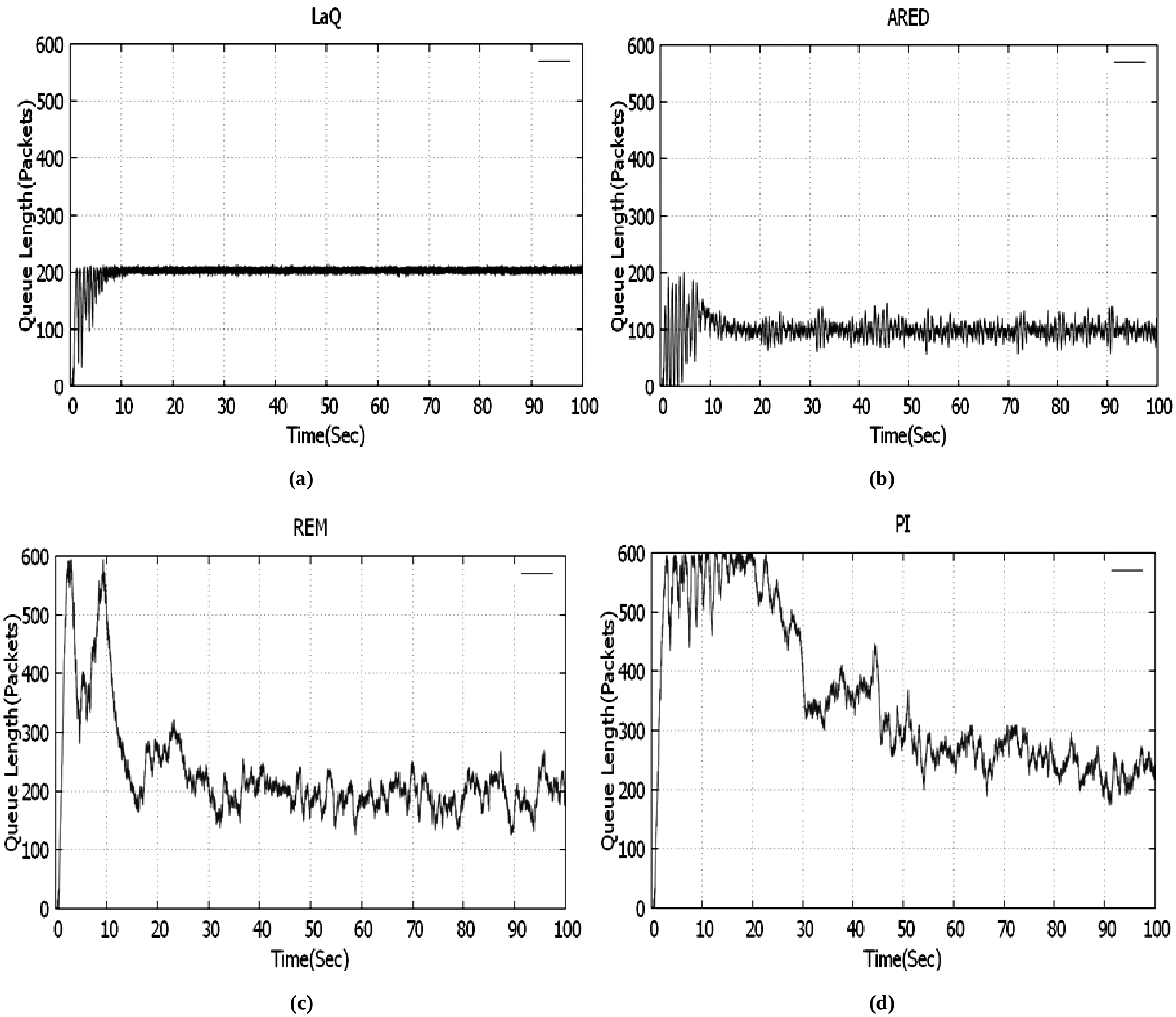


Figure 7: Queue length under random start time of (a) LaQ (b) ARED (c) REM (d) PI

effect of random starting time of each TCP connection are analyzed on the stability of AQM algorithms. From Figure 7, it is observed that the proposed algorithm is more stable than others under different starting time. It takes around 6 sec to bring the queue length into steady state which is very less value than others. The ARED algorithm manages to control queue length at 100 packets but not at given target (200 packets) with setting time 12 sec. The settling time of REM and PI are more than 30 sec which is quiet higher than LaQ. It shows that the proposed algorithm efficiently control the queue length around target under different starting time of TCP flows as compared to other existing algorithms.

4.1.4. Experiment 4: Stability under multi bottleneck links

In this experiment, the stability of AQM algorithms are analyzed under multi bottleneck links. For analysis we have consider a topology as shown in Figure 8. In this topology multiple bottleneck links are created. The first one is between router R2 and R3 and another is between router R4 and R5. Each link in between routers has the capacity 20Mb and delay is 20ms. The capacity and delay of other links are 100Mb and 10ms, respectively. Three different traffic sets, i.e. set-1, set-2 and set-3 are created with each of 200 TCP connections. The TCP connections of set-1 traverse through all the routers from R1 to R6 and the TCP

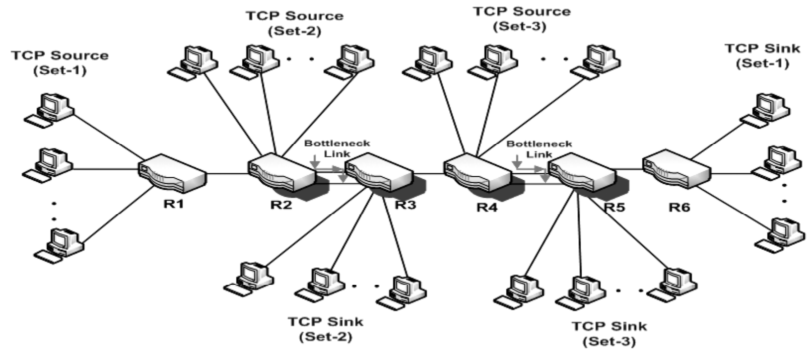
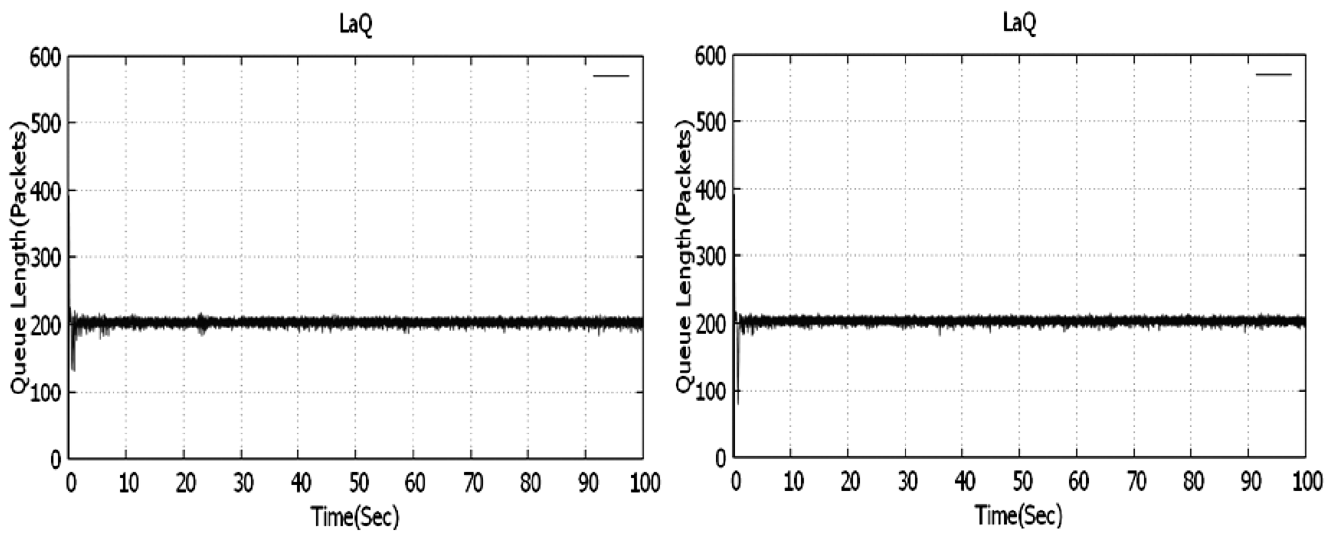


Figure 8: The multiple bottleneck network topology.

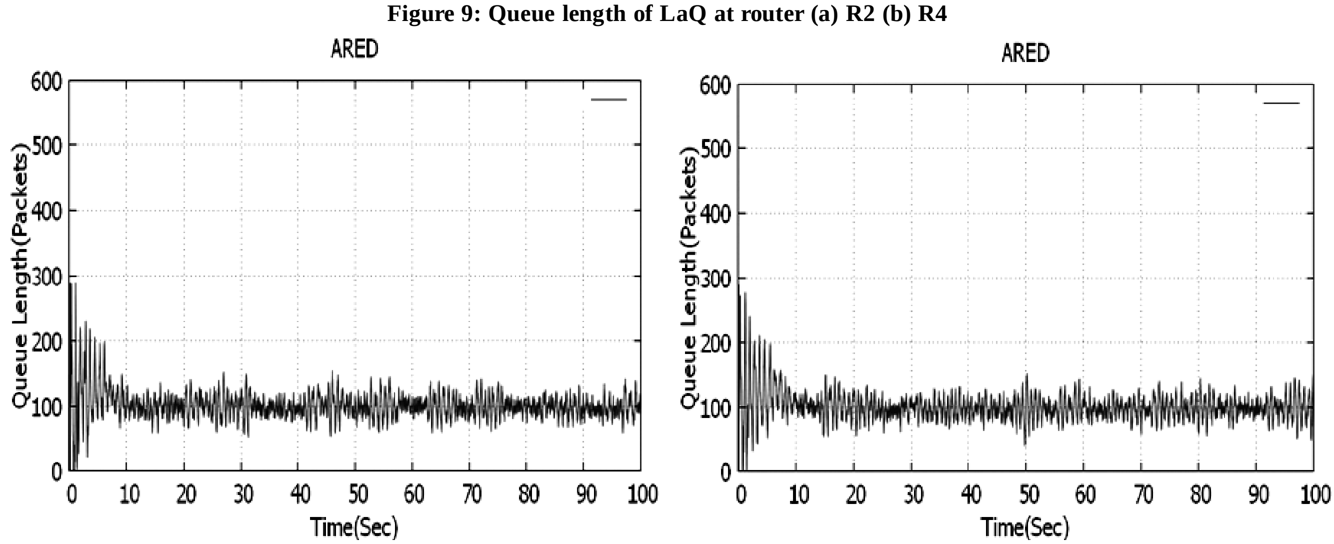
connections of set-2 traverse through the bottleneck link between R2 and R3 and the TCP connections of set-3traverse through the link R4 and R5. Each TCP connection in set-1 has same round trip time (RTT) of 240ms and RTT of each connection of set-2 and set-3 has same RTT of 80ms.

The queue length dynamics of each AQM controller are shown from Figure 9-12. It is observed that LaQ is more efficient in stabilizing the queue length around target at both the routers R2 and R4



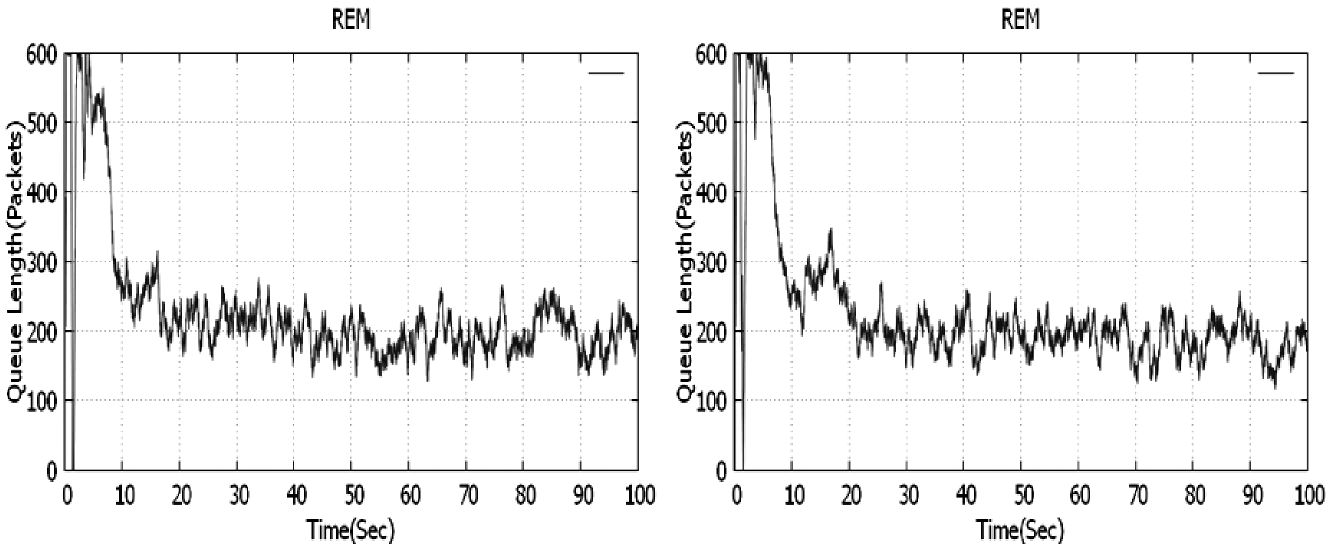
(a) **(b)**

Figure 9: Queue length of LaQ at router (a) R2 (b) R4



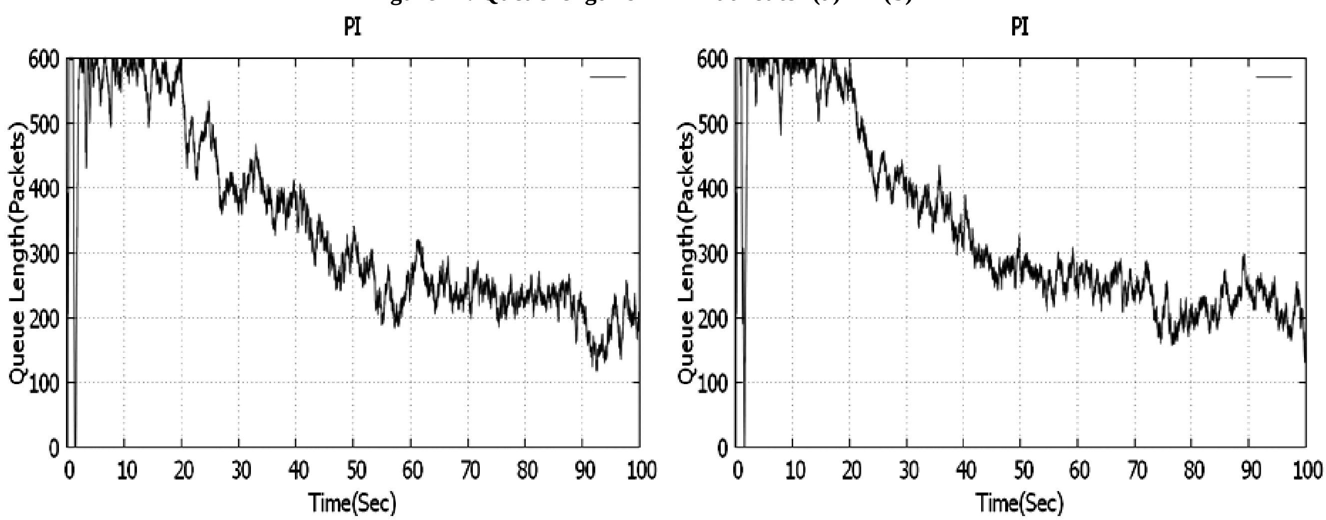
(a) **(b)**

Figure 10: Queue length of ARED at router (a) R2 (b) R4



(a) (b)

Figure 11: Queue length of REM at router (a) R2 (b) R4



(a) (b)

Figure 12: Queue length of PI at router (a) R2 (b) R4

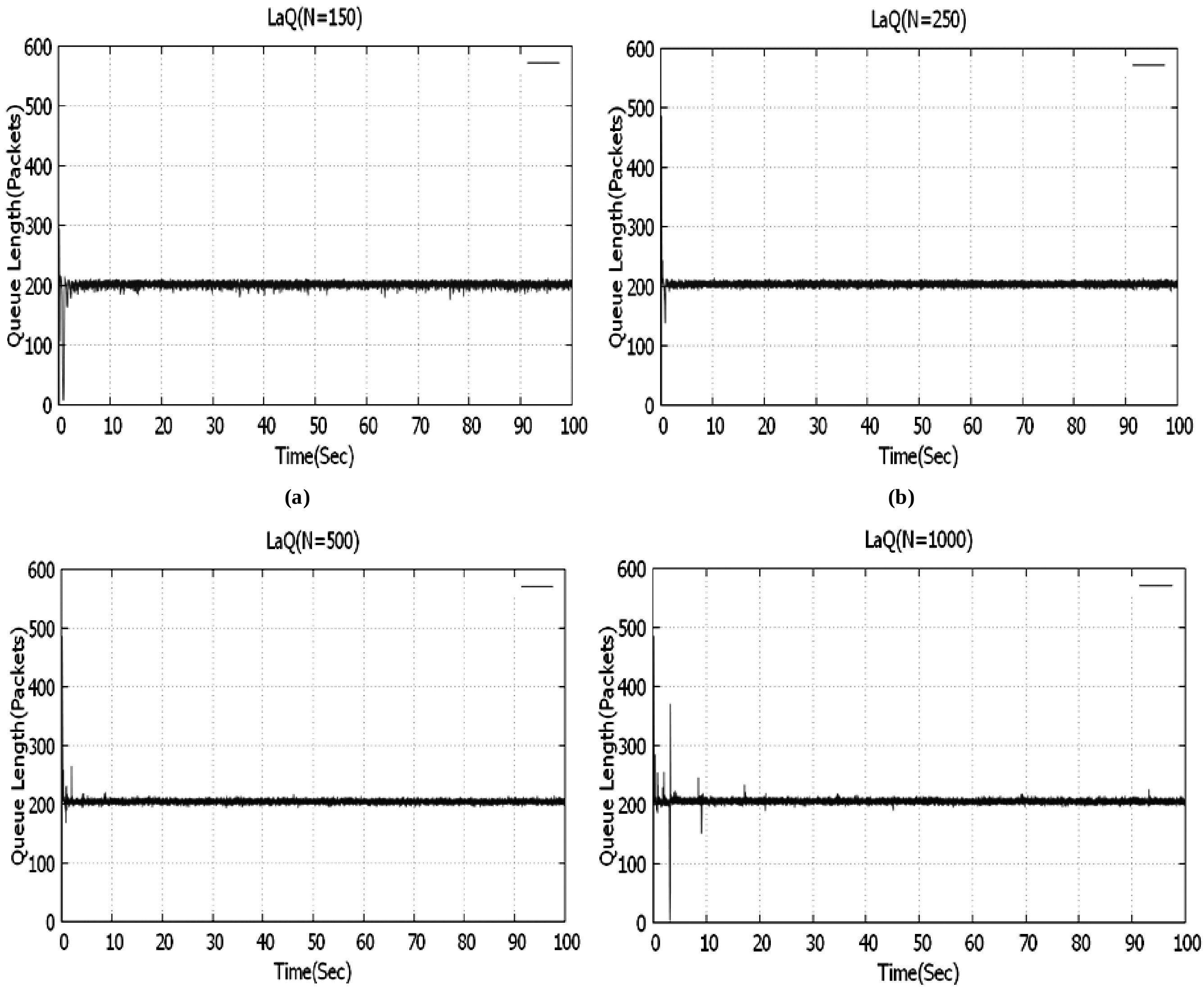
as compared to others. It shows that in presence of multiple bottlenecks the stability of proposed algorithm is maintained. The proposed algorithm achieve better stability with faster settling time than others in multi bottleneck links. Moreover, router R4 is more efficient than R2 in stabilizing the queue length.

4.2. Stability under different network parameters

In this section we explicitly analyze the stability and robustness of LaQ algorithm by varying the different network parameters such as number of connections, bottleneck bandwidth and RTT. We also analyze the stability of the system under different target queue length and heterogeneous delay. The stability of the algorithm is analyzed through the queue length measured.

4.2.1. Experiment 5: Varying the TCP connections

In this section, the stability of LaQ algorithm is analyzed with different TCP connections. In this the effect of light and heavy congestion on proposed algorithm is analyzed by changing the number of TCP connections. To achieve better stability of queue length in networks the amplitude of queue length fluctuation should be



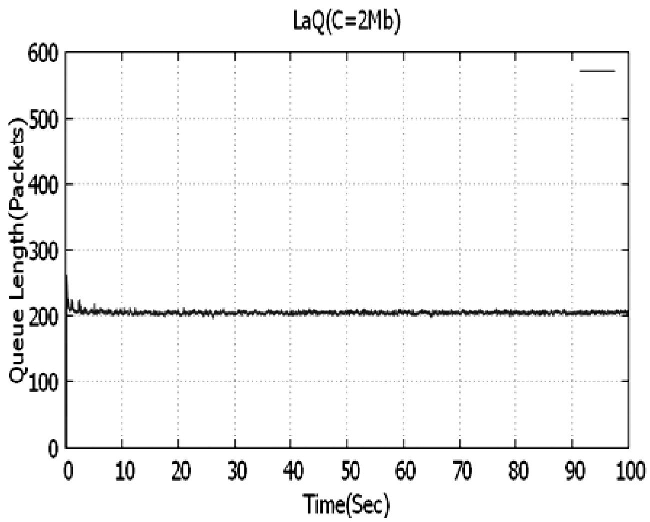
(a) (b) (c) (d)

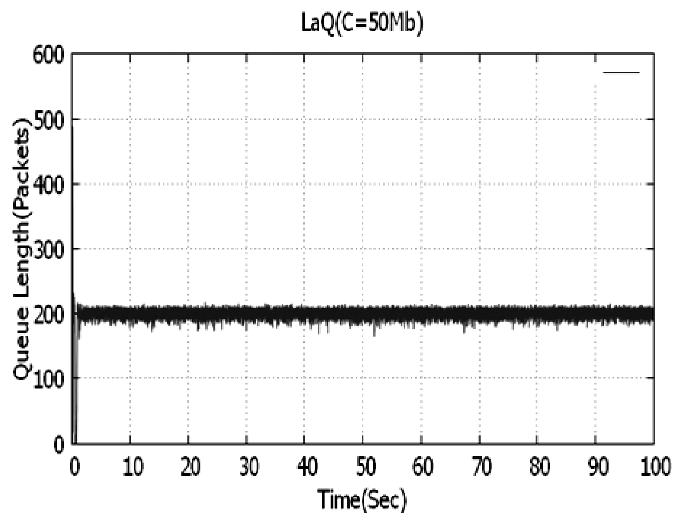
Figure 13: Queue length of LaQ for TCP connections
(a) N=150 (b) N=250 (c) N=500 (d) N=1000.

low. Large fluctuation in queue length brings high packet drop rate and leads to poor throughput. The queue length dynamics of LaQ is shown in Figure 13 for TCP connections N=150, N=250, N=500 and N=1000. From Figure 13 it is observed that proposed algorithm effectively stabilizes the queue length around the target under different TCP connections. Under heavy congestion N=1000 there is a small fluctuation of queue length for short period of time. However, LaQ achieves faster response under all the conditions (light and heavy congestion) with small deviation in queue length. It shows the robustness of LaQ under light and heavy congestion.

4.2.2. Experiment 6: Varying bottleneck link capacities

In this section the performance of LaQ algorithm is analyzed considering different bottleneck link capacities. We set the capacity of the bottleneck link R1 to R2 to 2Mb and 50Mb using the topology shown in Figure 4. The queue length dynamics of LaQ is shown in Figure 14 for 250 TCP connections. As shown in figure there is small overshoot in the queue length for both the scenario. The proposed algorithm LaQ has small deviation in queue length for varying link capacities.



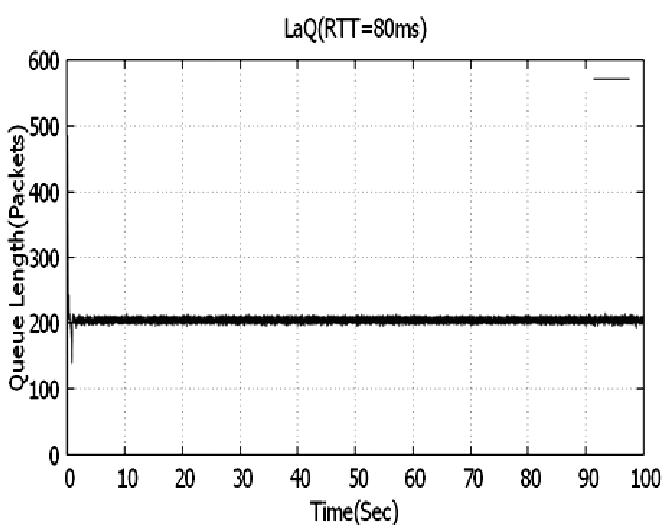


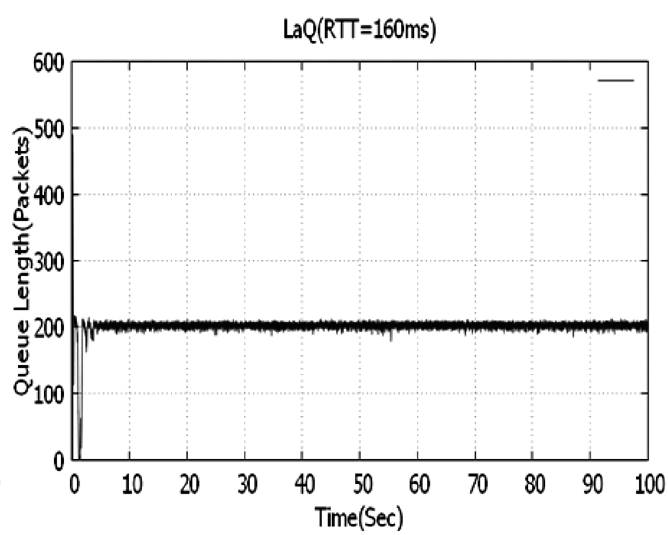
(a) **(b)**

Figure 14: Queue length of LaQ for bottleneck link capacity (C) (a) 2Mb (b) 50Mb.

4.2.3. Experiment 7: Varying Round Trip Times (RTTs)

In this experiment the performance of LaQ is analyzed considering 250 TCP connections with different round trip times. In this experiment two simulation scenarios have been created. In the first scenario, the RTT of each connection is 80ms and in second scenario the RTT is 160ms. Figure 15 demonstrates the queue length of LaQ for RTTs 80ms and 160ms. The settling time of LaQ is slightly higher at RTT 160ms than 80ms. The deviation of queue length is almost same after 4sec. It shows that LaQ algorithm effectively stabilizes the queue length around target under different RTTs.



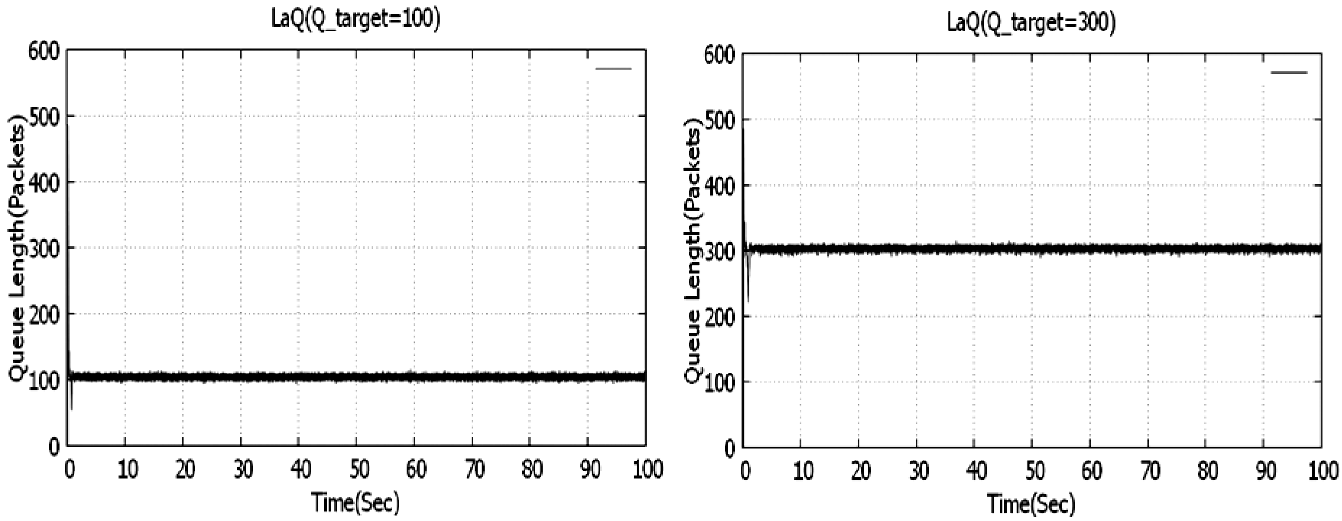


(a) **(b)**

Figure 15: Queue length of LaQ for round trip time (a) 80ms (b) 160ms.

4.2.4. Experiment 8: Varying target queue length

In this experiment the stability of LaQ is analyzed considering different target queue length. We set the target queue length to 100 and 300 packets. The queue length of LaQ is shown in Figure 16 for 250 TCP connections. It shows that the proposed algorithm efficiently stabilizes the queue length around arbitrary chosen target with small overshoots.

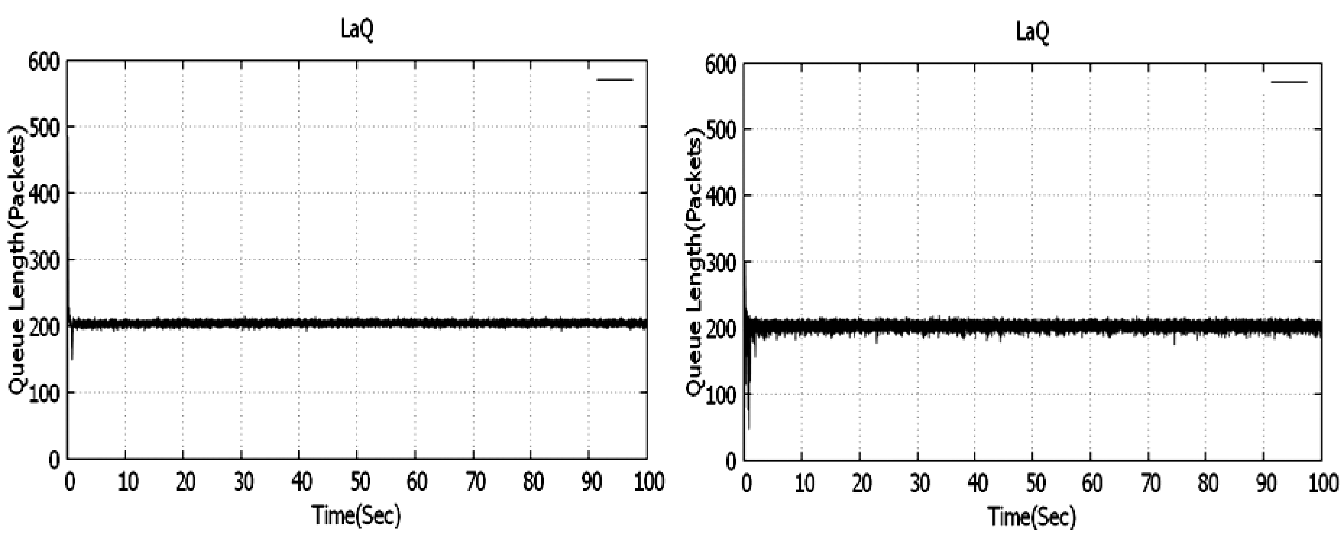


(a) **(b)**

Figure 16: Queue length of LaQ for target queue length (a) =100 (b) =300.

4.2.5. Experiment 9: Stability under different scenarios

In this experiment the stability of LaQ algorithm is analyzed under two different scenarios: (1) arrival of new traffic suddenly whose starting time is random (2) different RTT for different sets of TCP connections. In first scenario, 250 TCP connections created at the beginning of the simulation and then another 100 TCP connections are added suddenly whose starting time is random between 0 to 50 sec. Figure 17(a) shows the queue length of LaQ for the first scenario. It is observed from the figure that, LaQ is able to manage the queue length efficiently with faster response and lower deviation in queue length under first scenario. It shows that the proposed algorithm is robust under sudden arrival of TCP traffic whose start time is random. In second scenario, three different sets each of 100 TCP flows are created whose RTT is different from each other. The first set of TCP has RTT 80ms and second set has 120ms and third set has 160ms. The queue length of LaQ is shown in Figure 17(b). It is observed that LaQ maintain its queue length efficiently around target with small response time. There is small deviation in queue length at the beginning and then it is able stabilize around given target.



(a) **(b)**

Figure 17: Queue length of LaQ under (a) first scenario (b) second scenario.

5. CONCLUSIONS

In this paper a novel AQM algorithm called LaQ is proposed based on packet loss ratio and current queue length to control congestion efficiently. It brings stability in the system by controlling the resources of the router properly. During congestion, LaQ adjust the packet drop probability properly to control the router queue. The packet drop probability of proposed algorithm is adjusted based on measured packet loss ratio value and current queue length dynamically. We use control theory to analyze the stability of proposed system and validated using NS2 simulator. The proposed algorithm LaQ is compared with ARED, REM and PI in single bottleneck and multiple bottleneck link topology.

The result shows that the proposed algorithm efficiently stabilizes the queue length around target with faster response and robust under significant change of network parameters such as TCP connections, bottleneck bandwidth, RTT, arbitrary target queue length. The proposed algorithm outperforms existing algorithms with respect to throughput, link utilization, average delay and packet loss ratio. In addition, LaQ maintain its queue length closer to the target than others.

REFERENCES

[1] B. Braden et al. RFC 2309, “Recommendations on queue management and congestion avoidance in the internet”, 1998.

[2] S. Floyd, and V. Jacobson, “Random early detection gateways for congestion avoidance”, IEEE/ACM Trans on Netw. Vol. 1, No.4, pp. 397–413, 1993.

[3] S. Floyd, R. Gummadi, and S. Shenker, “Adaptive RED: an algorithm for increasing the robustness of RED’s active queue management”, Technical report. 2001.

[4] T.H. Kim, and K.H. Lee, “Refined adaptive RED in TCP/IP networks”, SICE-ICASE 2006, Bexco, Busan, Korea, pp. 3722–3725, 2006.

[5] K. Zhou, K.L.Yeung, O. Victor, and K. Li, “Nonlinear RED: A simple yet efficient active queue management scheme”, Elsevier Journal of Computer Networks, Vol.50, pp. 3784-3794, 2006.

[6] T.J. Ott, T.V. Lakshman, and L. Wong, “SRED: Stabilized RED”, IEEE INFOCOM 1999, Vol. 3, March 21–25, New York, pp.1346-1355, 1999.

[7] M.P. Tahiliani, K.C. Shet, and T.G. Basavaraju, “CARED: Cautious Adaptive RED gateways for TCP/IP networks”, Journal of Network and Computer Applications, Vol. 35, pp. 857-864, 2012.

[8] C. Wang, J. Liu, B. Li, K. Sohraby, and Y. T. Hou, “LRED: a robust and responsive AQM controller using packet loss ratio measurement”, IEEE Transactions on Parallel and Distributed Systems, Vol.18, pp. 29-43, 2007.

[9] S. Liu, T. Basar, and R. Srikant, “Exponential-RED: a stabilizing AQM scheme for low- and high-speed TCP protocols”, IEEE/ACM Transactions on Networking, Vol. 13, pp.1068-1081, 2005.

[10] J. Aweya, M. Ouellette, and D.Y. Montuno, “A control theoretic approach to active queue management”, Computer Networks, Vol. 36, pp. 203-35, 2001.

[11] S. Kunniyur, and R. Srikant, “Analysis and design of an adaptive virtual queue (AVQ) controller for active queue management”, ACM SIGCOMM 2001,ACM New York, NY, USA, pp.123-134, 2001.

[12] 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.

[13] A. Alsaaidah, M. Zalisham, M. Fadzli, and H. A. Jaber, “Gentle-BLUE: A New Method for Active Queue Management”, Advanced Computer Science Applications and Technologies (ACSAT) 3rd International Conference, pp. 67-72, 2014.

[14] B. Wydrowski, and M. Zukerman, “GREEN: an active queue management controller for a self managed internet”, ICC 2002, Vol. 4, pp. 2368-2372, 2002.

[15] C. Hollot, V. Misra, D. Towsley, and W. Gong, “A control theoretic analysis of RED”, EEE INFOCOM 2001; Vol. 3, Anchorage, AK, USA, pp. 1510-1519, 2001.

[16] J. Sun, G. Chen, K. Ko, S. Chan, and M. Zukerman, “PD-controller: a new active queue management scheme”, IEEE GLOBECOM 2003, Vol. 6, pp. 3103–3107, 2003.

[17] F. Yanfie, R. Fengyuan, and L. Chuang, “Design a PID controller for active queue management”, IEEE Symposium on Computers and Communications ISCC 2003, Vol. 2, Kemer-Antalya, Turkey, pp. 985–990, 2003.

[18] S. Athuraliya, S. Low, V. Li, and Q. Yin, " REM: active queue management", IEEE Network, Vol. 15, No. 3, pp. 48–53, 2001.

[19] J. Sun, and M. Zukerman, "RaQ: a robust active queue management scheme based on rate and queue length", Comput. Commun., Vol. 30, pp.1731-1741, 2007.

[20] E. C. Park, H. Lim, K. J. Park, and C. H. Choi, "Analysis and design of the virtual rate control controller for stabilizing queues in TCP networks", Computer Networks, Vol. 44, pp. 17-41, 2004.

[21] C. Long, X. Guan, B. Zhao, and J. Yang, "The Yellow active queue management controller", Computer Networks, Vol. 47, No. 4, pp.525-550, 2005.

[22] K. Nichols, and V. Jacobson, "Controllig Queue Delay", Communications of the ACM Vol. 55 No. 11, pp.42-50, 2013.

[23] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg, "PIE: A Lightweight Control Scheme to Address the Bufferbloat Problem", IEEE14th International Conference on High Performance Switching and Routing, pp.148-155, 2013.

[24] R. Adams, "Active queue management: a survey", IEEE Communications Surveys & Tutorials, Vol. 15, No. 3, pp.1425–1476, 2013.

[25] Q. Xu, and J. Sun, "A simple active queue management based on the prediction of the packet arrival rate", Journal of Network and Computer Applications, Vol. 42, pp.12-20, 2014.

[26] Y. Ariba, F. Gouaisbaut, and Y. Labit, "Feedback Control for Router Management and TCP/IP Network Stability", IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, Vol. 6, No. 4, pp. 255-266, 2009.

[27] L. Wang, L. Cai, X. Liu, X. Shen, and J. Zhang, "Stability analysis of multiple-bottleneck networks", Computer Networks, Vol. 53, pp. 338-352, 2009.

[28] 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", SIGCOMM Comput. Commun. Rev., Vol. 30, No. 4, pp. 151-160, 2000.

[29] T. Issariyakul, and E. Hossain, "Introduction to Network Simulator NS2", 1st ed., Springer Publishing Company, Incorporated, 2008.