Modified RED Algorithm to Improve The Performance of Web Traffic

Maulik K. Dadhania
Department of Computer Science and Engineering
National Institute of Technology Karnataka,
Surathkal, Karnataka, India 575025
e-mail: maulik.dadhania@gmail.com

K. Vinay Kumar
Department of Computer Science and Engineering
National Institute of Technology Karnataka,
Surathkal, Karnataka, India 575025
e-mail: vinay@nitk.ac.in

***Abstract*—Reducing the delay of web traffic has been an important issue in Internetworks. However, it is difficult to reduce delay while improving throughput. This is because, in order to reduce delay queue size should be less but to improve throughput queue size should be more. In our modification, web traffic and ECN marked packets are dropped only when high congestion occurs at router. We propose a solution to improve response time as well as the number of packets transmitted of web traffic without affecting throughput of bottleneck link. We applied this solution to RED algorithm and tested the performance and efficiency of modified RED algorithm as compared to the original RED algorithm.**

Keywords-RED; ECN; AQM; Queue Size;

I. INTRODUCTION

The traditional router queue management approach sets the router queue size as maximum queue size for each queue. Router can accept packets for the queue until the queue is full thereafter it drops subsequent incoming packets. This technique is known as tail-drop. It has two important drawbacks [1]:

- *Lock-Out:* It may happen that, tail-drop allows a single flow or few flows to occupy queue space hence preventing other flows from getting space in the queue.
- *Full-Queues:* End to end delay of the packets is increased when queue is about to full or completely full.

Active queue management (AQM) is an approach to solve the full-queues problem [1]. It drops packets before the queue becomes full hence sender nodes can respond to congestion before the queue gets overflow. Random-drop and front-drop gives the solution to lock-out problem [1] because the chance of dropping packets from same flow is lesser in these techniques.

In this paper, we have compared the performance of web traffic and elephant traffic with traditional packet drop and modified packet drop techniques using random early detection (RED) and modified RED.

The remaining section of this paper is organized as follows. In section II, we discuss the original RED algorithm. We present the modified algorithms in section III. We provide detail of experimental environment in section IV. In section V, we discuss the experimental results. Finally, we conclude the paper in section VI.

II. RANDOM EARLY DETECTION (RED)

RED [2] is a most popular and widely deployed active queue management algorithm. RED tracks the average queue size based on low pass filter and drops\marks packets to control the congestion based on packet drop probabilities. All incoming packets are accepted by RED when buffer is almost empty. The probability of dropping for incoming packets increases with growing queue size.

RED queue has the following parameters:

- The average queue size (q_{avg}): q_{avg} is used to identify the average size of the queue. The queue size can be in bytes or in packets.
- Maximum Threshold (q_{max}): q_{max} is used to calculate the packet loss ratio. When the average queue exceeds this value, this packet must be discarded.
- Minimum threshold (q_{min}): q_{min} is used to calculate the packet loss ratio. When the average queue is below this value, the data packets must not be discarded.
- Loss rate (p_d): p_d is the probability that the currently arrived packet is discarded.
- Maximum drop probability (p_{max}): p_{max} is the maximum probability value of p_d. If the p_d value exceeds this threshold, the packets must be discarded.
- Weights (w_q): w_q is used to calculate the average queue size.

RED queue has the following variables:

- Instantaneous queue size: q
- Number of packets that arrived since the last packet drop: count

When, the average queue size reaches the minimum threshold, RED starts dropping packets with probability, in this way RED avoids the global synchronization problem. When the average queue size reaches maximum threshold, RED drops all arriving packets using tail-drop strategy until average queue size becomes less than the maximum threshold.

The RED algorithm works in two parts: estimation of the average queue size and the decision regarding dropping of an incoming packet.

978-0-7695-4941-5/13 $26.00 © 2013 IEEE
DOI 10.1109/ACCT.2013.45

1. *Estimation of the average queue size for each packet arrival:*
   $q_{avg(i)} = (1 - w_q) * q_{avg(i-1)} + q * w_q$

2. *Packet drop probability:*
   $p_d = 0$ if $q_{avg} < q_{min}$
   $= (q_{avg}-q_{min})/(q_{max}-q_{min})*p_{max}$ if $q_{min} \leq q_{avg} < q_{max}$
   $= 1$ if $q_{avg} \geq q_{max}$

Algorithm 1: RED Algorithm

```
For arrival of every packet:
if q_avg <= q_min then
        Enqueue the packet
end
if q_avg >= q_max then
    Drop the packet
end
if q_min < q_avg < q_max then
    Mark or drop the packet from the queue with
    certain probability p_d
end
```

AQM algorithms do not differentiate mice and elephant traffic. In this paper, we have modified RED algorithm to reduce the queue delay experienced by mice traffic. Also, we have compared the performance of ECN, front-drop, tail-drop, random-drop and modified algorithm with respect to how it affects the web traffic and elephant traffic.

III. PROPOSED MODIFICATION OF ALGORITHMS

In general, we observe two problems in AQM algorithms which can be described as follows:

1. AQM algorithms consider all packets from different traffic as identical. Web traffic is short-lived flow and considered as mice traffic. Short-lived flows have small congestion window size. Hence, dropping the packets of short-lived flow does not reduce the congestion much [3].

2. Router informs congestion to sender by dropping packets or by marking explicit congestion mechanism (ECN) bit in IP-header of the packet [8]. If ECN marked packet is selected for early-drop, sender learns the congestion at router somewhat late which delays the sender to set its window size to appropriate rate.

To solve these problems we drop packets of long-lived flows instead of short-lived flows. To differentiate web traffic from other traffics at router, we have used a flag 'Mice' in option field of IP-header. Network layer of sender sets the Mice flag to one for web traffic. Elephant traffic has Mice flag value zero. Modified algorithm can drop web traffic packets only when there is high congestion i.e. when average queue size is greater than maximum threshold. Until the high congestion, early packet drops are selected from elephant traffic.

Whenever mice or ECN packet is selected as early-drop, at that time elephant traffic packet is dropped in place of mice or ECN. Hence, sender of elephant traffic responds to congestion little faster and mice sender does not reduce the congestion window. This leads to reduced packet loss rate and reduced delay for web traffic. There is no extra overhead to differentiate packet at router.

This technique can be applied in any AQM algorithm. In this paper, we have modified RED algorithm. We have also modified front-drop and random-drop. We cannot modify tail-drop because it has extra overhead to find packet which is not marked for ECN and Mice.

Algorithm 2: Modified RED Algorithm

```
For arrival of every packet:
if q_avg <= q_min then
        Enqueue the packet
end
if q_avg >= q_max then
    Drop the packet
end
if q_min < q_avg < q_max then
    if ECN == 1 or Mice == 1 then
            Enqueue the packet
    end
    Mark or drop the elephant packet from the queue
    with certain probability p_d
end
```

A. Modified Front-Drop

Front-drop can drop\mark ECN packets from the front of queue during early-drop or when queue is full. Modified front-drop algorithm does not drop the packets that are marked as mice or ECN in early-drop. It starts search from rear until the packet that is not marked as mice or ECN. It can drop mice or ECN marked packet only after average queue size is more than maximum threshold. The overhead of searching elephant packet is very less because elephant traffic occupies the maximum room in queue.

B. Modified Random-Drop

Random-drop can randomly select the packet for drop or mark ECN in early-drop. It may happen that randomly selected packet is already ECN marked or form web traffic. Hence, this algorithm tries to select packet which is not ECN marked or not from web traffic. The algorithm attempts to select the random packet maximum three times. After three trials, if algorithm could not find a packet which is not ECN marked or not from web traffic, the third selected packet is dropped.

IV. EXPERIMENTAL ENVIRONMENT

To verify the effectiveness of the modified algorithm, we have used network simulator NS2 to simulate it [9]. The experimental topology is a typical single bottleneck dumbbell topology as shown in Fig 1. [4]:

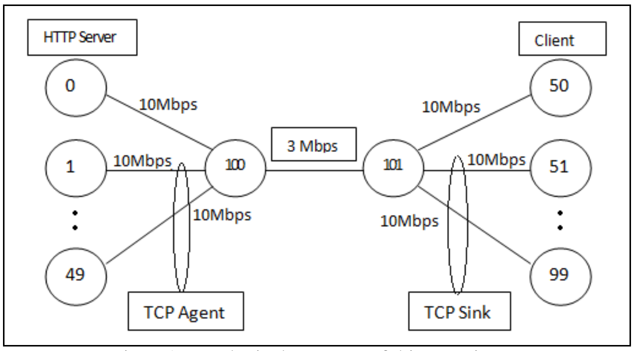

Figure1. Topological structure of this experiment

Table I describes the experimental parameters used in topology [5] [7].

TABLE I. EXPERIMENTAL PARAMETERS

Parameters	***Values***
NS version	2.35
Duration	110 seconds
FTP start time	0
FTP end time	110
FTP packet size	1024
TCP Type	SACK
HTTP start time	10
HTTP stop time	110
Bottleneck Queue length	100 packets
No. of FTP flows	49
Queue Weight (w_q)	0.002
Maximum drop probability (p_{max})	0.1
Minimum threshold (q_{min})	5
Maximum Threshold (q_{max})	$3 * q_{min} = 15$

V. EXPERIMENTAL RESULTS

We demonstrate the performance improvement of modified RED based on end-to-end delay, average and instantaneous queue size, number of successfully transmitted packets of web traffic, percentage of packet drop and link utilization of bottleneck link.

A. Comparison of Average and Instantaneous Queue Size

1) Front-Drop (Fig. 2)

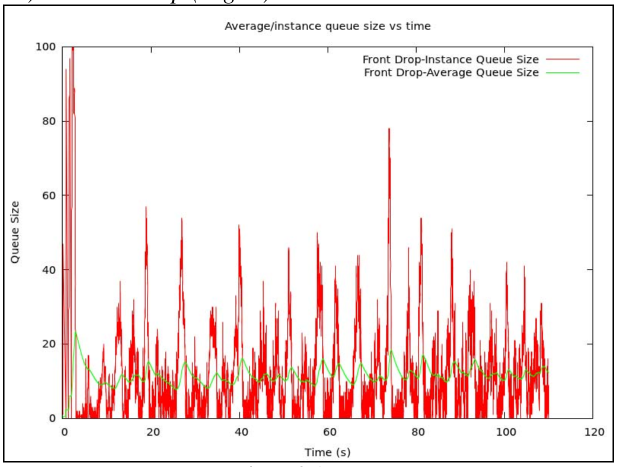

Figure 2.A: RED

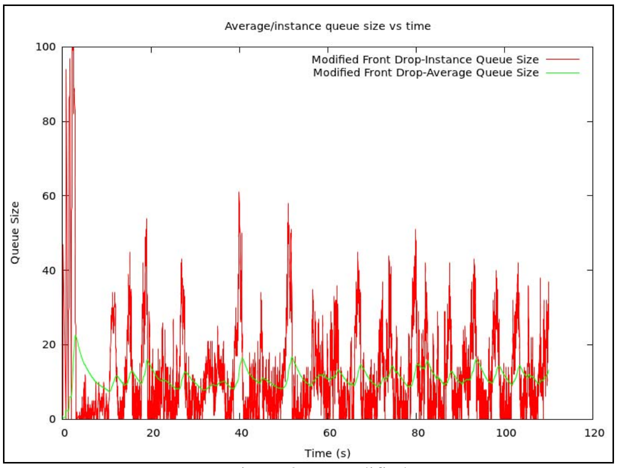

Figure 2.B: Modified RED

Figure 2: Average and Instantaneous Queue Size using Front-Drop

2) Front-Drop with ECN (Fig. 3)

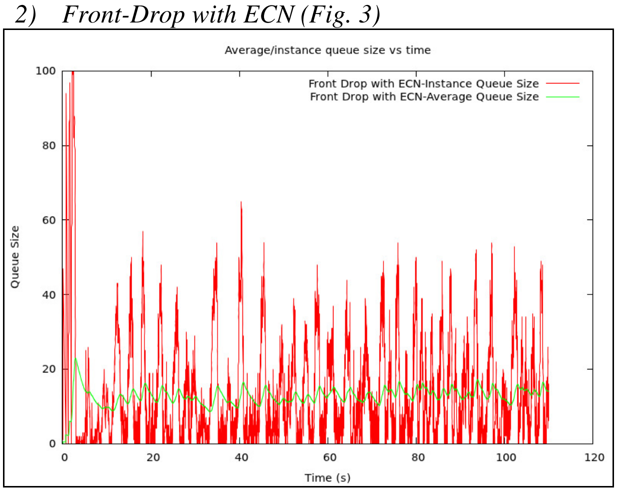

Figure 3.A: RED

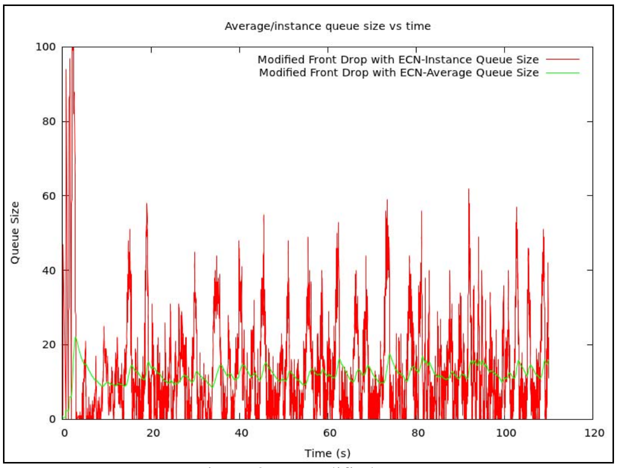

Figure 3.B: Modified RED

Figure 3: Average and Instantaneous Queue Size using Front-Drop with ECN

3) Random-Drop (Fig. 4)

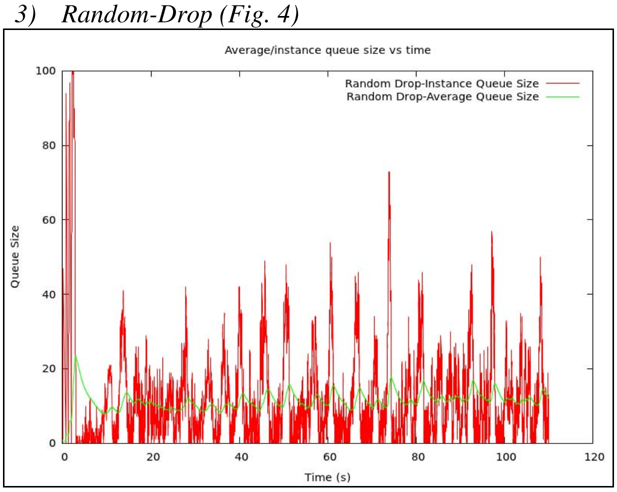

Figure 4.A: RED

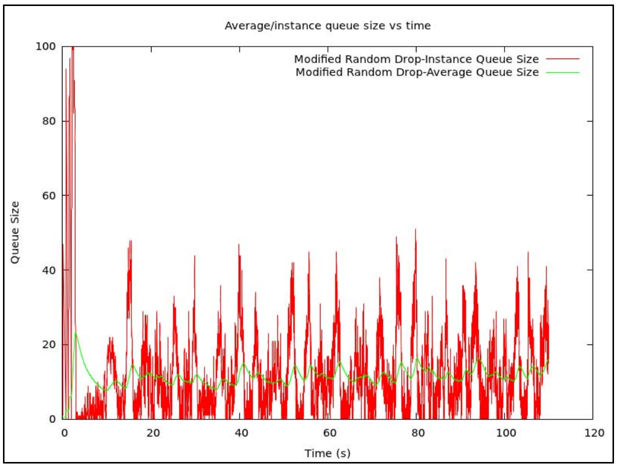

Figure 4.B: Modified RED

Figure 4: Average and Instantaneous Queue Size using Random-Drop

4) Random-Drop with ECN (Fig. 5)

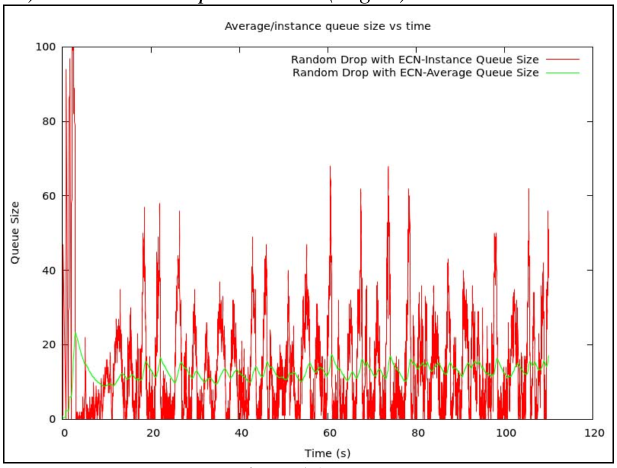

Figure 5.A: RED

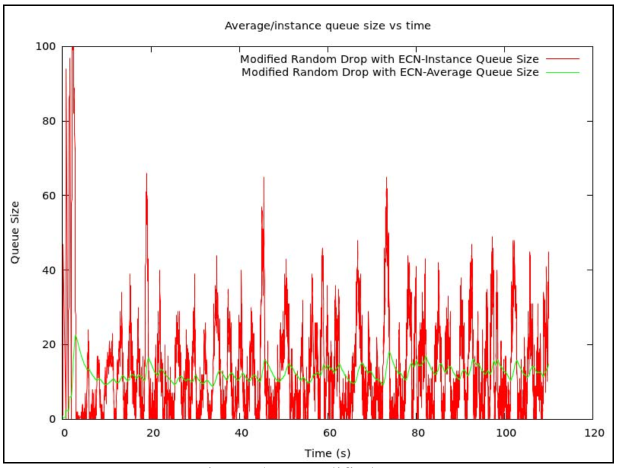

Figure 5.B: Modified RED

Figure 5: Average and Instantaneous Queue Size using Random-Drop with ECN

5) Tail-drop (Fig. 6)

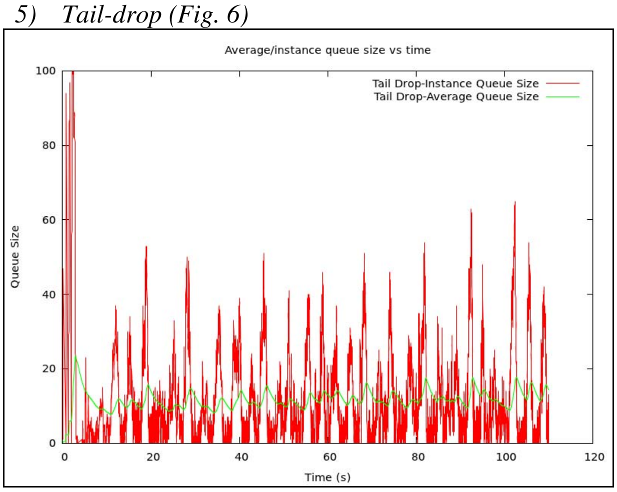

Figure 6.A: RED

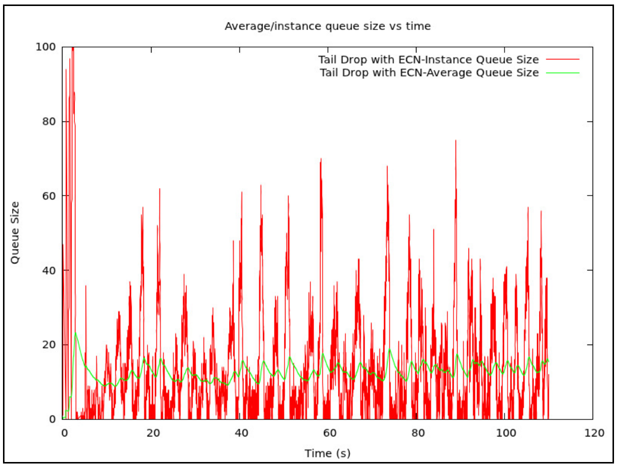

Figure 6.B: RED with ECN

Figure 6: Average and Instantaneous Queue Size using Tail-Drop

Average queue size depends on the instantaneous queue and previous average queue size. Peak values of instantaneous queue size affect more in average queue size. Modified techniques reduce the highest peak of instantaneous queue size and make the average queue size more stable. From Fig. 2 to 6, we observe that tail-drop has more variation in instantaneous queue size than front-drop and random-drop.

B. Comparison of Delay

Delay increases as average queue size increases. Modified algorithm reduces the pick value of average queue size therefore it reduces the pick value of delay which can be observed from the Fig. 7 to 11. These graphs have different scale on delay axis for modified and original algorithm.

1) Front-Drop (Fig. 7)

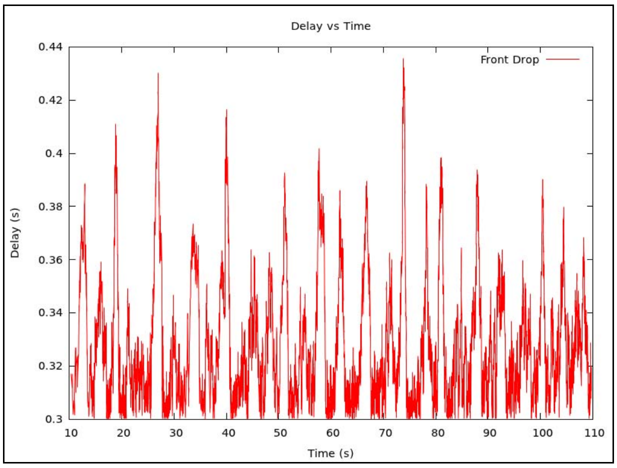

Figure 7.A: RED

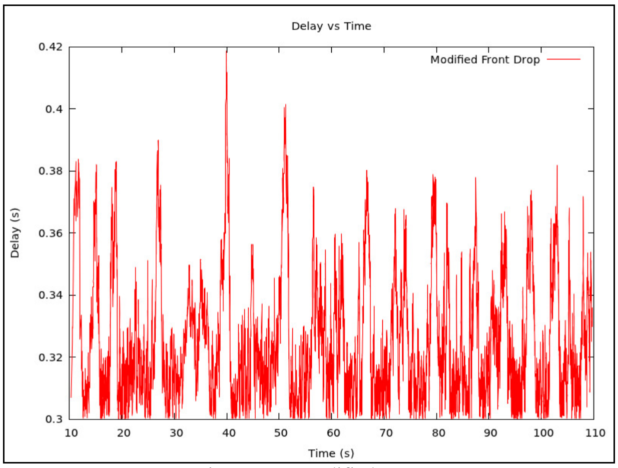

Figure 7.B: Modified RED

Figure 7: The Delay of web Traffic using Front-Drop

2) Front-Drop with ECN (Fig. 8)

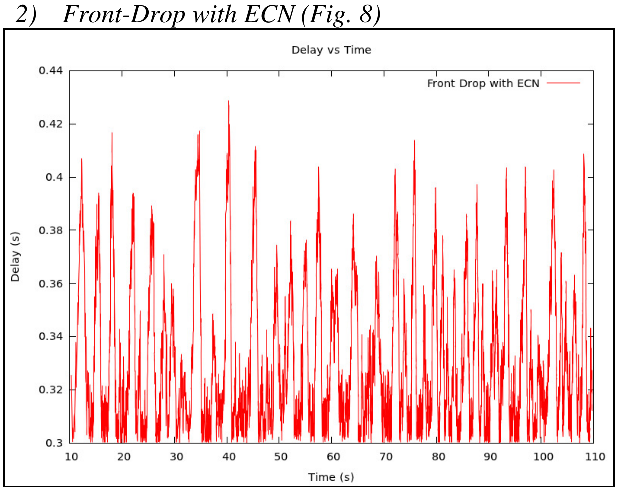

Figure 8.A: RED

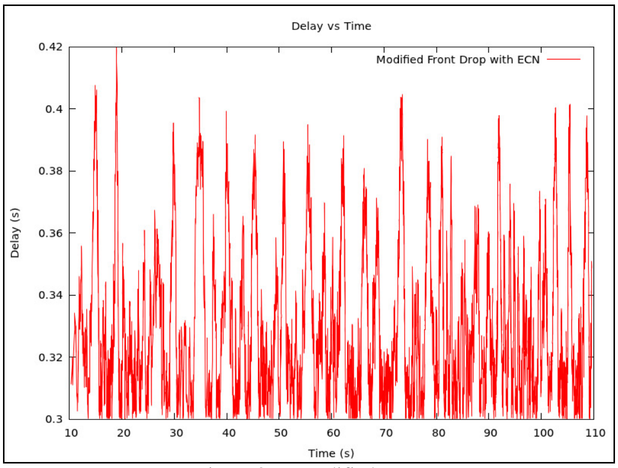

Figure 8.B: Modified RED

Figure 8: The Delay of web Traffic using Front-Drop with ECN

3) *Random-Drop (Fig. 9)*

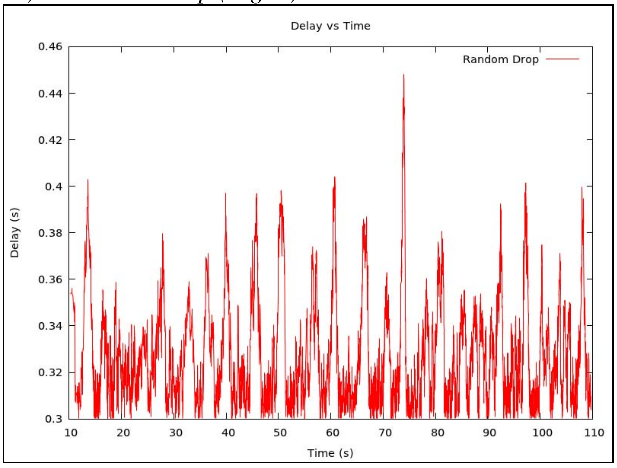

Figure 9.A: RED

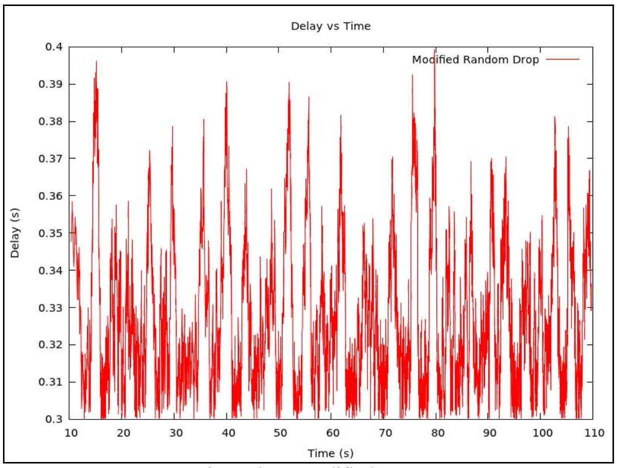

Figure 9.B: Modified RED

Figure 9: The Delay of web Traffic using Random-Drop

4) *Random-Drop with ECN (Fig. 10)*

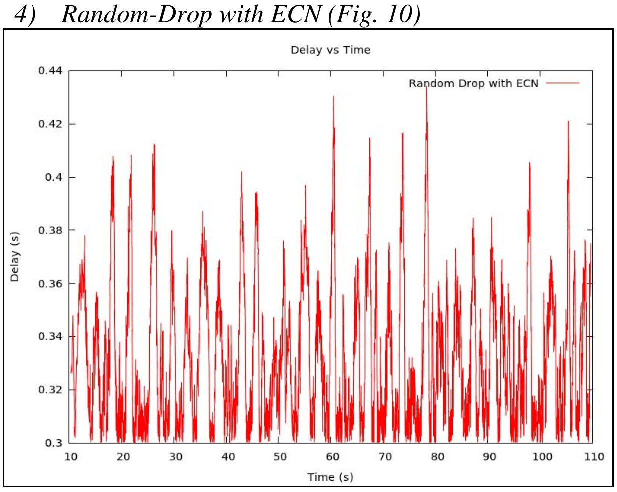

Figure 10.A: RED

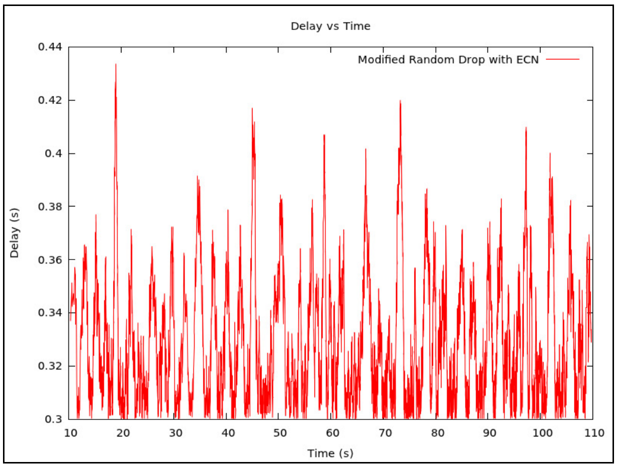

Figure 10.B: Modified RED

Figure 10: The Delay of web Traffic using Random-Drop with ECN

5) *Tail-drop with and without ECN (Fig. 11)*

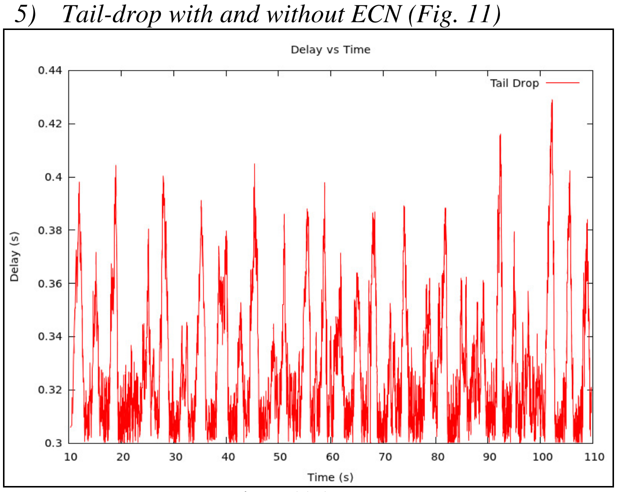

Figure 11.A: RED

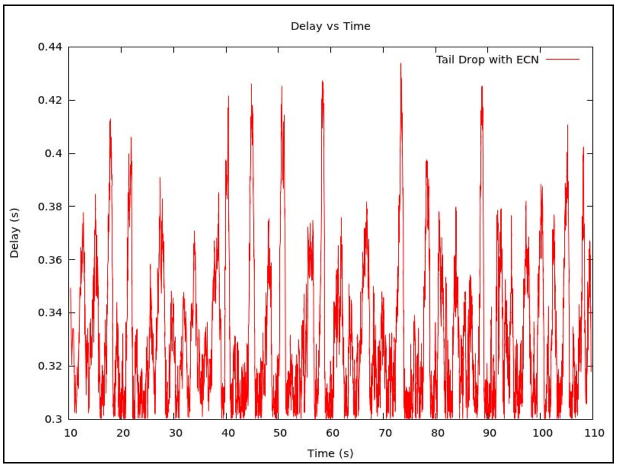

Figure 11.B: RED with ECN

Figure 11: The Delay of web Traffic using Tail-Drop

C. Number of successfully transmitted packets of web traffic (Fig. 12)

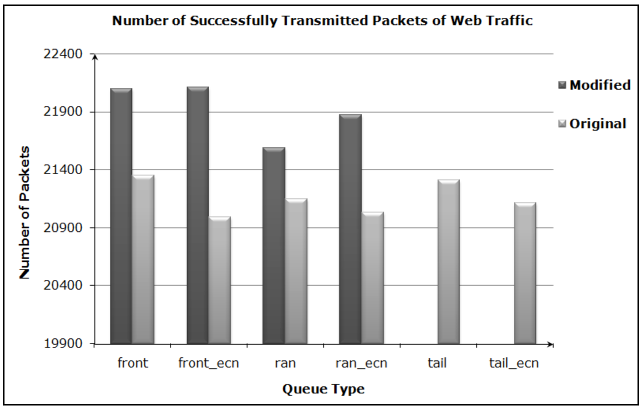

Figure 12: Number of Successfully Transmitted Packets of Web Traffic

Numbers of web traffic packets successfully transmitted are more in the case of modified RED algorithm than original RED algorithm. Numbers of packets sent by the http node are less when ECN is enabled, in original algorithm. When ECN is enabled, original drop techniques reduce the number of successful web traffic packets sent. Modified front-drop can transmits more web traffic packets then modified random-drop.

D. Average end-to-end delay of web traffic (Fig. 13)

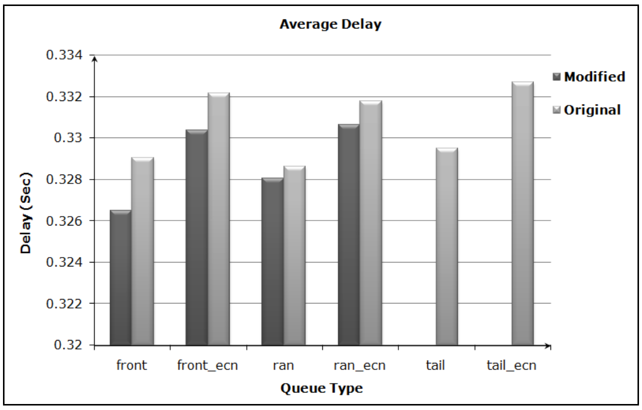

Figure 13: Average end-to-end Delay of Web Traffic

Here, we have considered delay only for mice traffic because elephant traffic is not delay sensitive. Modified techniques have less delay as compared to respective original techniques. Tail-drop has highest delay because each packet has to travel in queue from end to front.

E. Percentage of packet drop (Fig. 14)

Percentage of packet drop is reduced while ECN is enabled in original drop techniques. It is reduced further in modified techniques. Tail-drop has more packet drops then front-drop or random-drop because sender receives the congestion notification somewhat late in case of tail-drop.

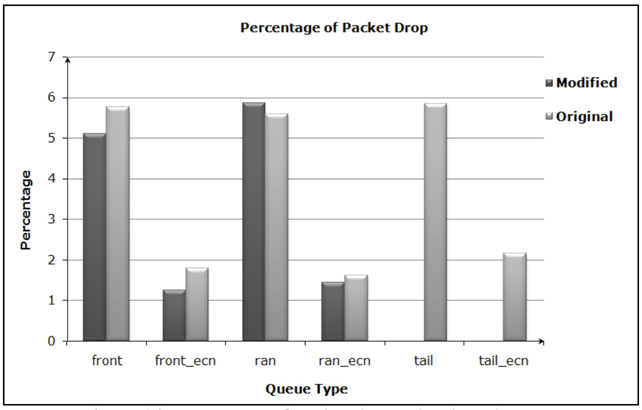

Figure 14: Percentage of packet drop at bottleneck router

F. Link utilization of bottleneck link (Fig. 15)

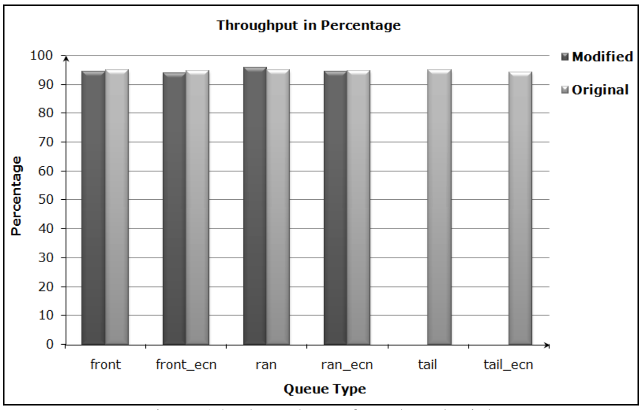

Figure 15: Throughput of Bottleneck Link

Throughput of modified and original techniques is almost same. This shows that the improvement of delay of web traffic does not affect the throughput.

VI. CONCLUSION AND FUTURE WORK

Our conclusions are based on modifications done to RED algorithm. Proposed method improves the response time of web traffic without affecting the throughput of bottleneck link. The peak value of the instantaneous queue size is reduced in modified algorithm. Hence, the average queue size becomes more stable. Because of this, the peak values of the queue delay are also reduced which lead to reduced average queue delay is reduced in modified algorithm.

These methods give little higher priority to web traffic than elephant traffic which improve the number of mice packets successfully transmitted. The number of packet drops in modified technique is less than any other technique when ECN is enabled. Hence, it reduces the retransmission overhead. These modifications give surpassed performance in any parameters when ECN is enabled. Front-drop and random-drop gives better performance then tail-drop. Front-drop responds to congestion faster than random-drop or tail-drop hence it reduces delay more than others.

This modification can be applicable in any AQM algorithm. We will modify Adaptive RED [6] algorithm and analyze the performance of RED and ARED in future work.

ACKNOWLEDGEMENT

The authors would like to thank Mr. Dipesh Raghuvanshi and Mr. Mohit P. Tahiliani for their help in performing the simulation experiments and making this paper reality.

REFERENCES

[1] B. Braden et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet", Internet Engineering Task Force, RFC 2309, April 1998.

[2] S. Floyd, V. Jacobson, "Random early detection gateways for congestion avoidance", in IEEE/ACM Transactions on Networking, vol. 1, issue 4, pp. 397–413, August 1993.

[3] L. Long, J. Aikat, K. Jeffay, and F. Smith., "Differential congestion notification: taming the elephants", in 12th IEEE International Conference on Network Protocols, pp. 118–128, October 2004.

[4] L. Long, J. Aikat, K. Jeffay, and F. Smith, "The Effects of Active Queue Management and Explicit Congestion Notification on Web Performance", in IEEE/ACM Transactions on Networking, vol. 15, issue 6, pp. 1217–1230, December 2007.

[5] S. Floyd, RED: Discussions of Setting Parameters, November 1997, "http://www.icir.org/floyd/REDparameters.txt".

[6] S. Floyd, R. Gummadi, S. Shenker, "Adaptive RED: an algorithm for increasing the robustness of RED's active queue management", in Technical report (http://www.icir.org), August 2001.

[7] M. Claypool, R. Kinicki, and M. Hartling, "Active Queue Management for Web Traffic" in IEEE International Conference on Performance, Computing, and Communications, pp. 531–538, 2004.

[8] J. Salim, N. Nortel, and U Ahmed, "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks", Internet Engineering Task Force, RFC 2884, July 2000.

[9] The Network Simulator - ns-2 "http://www.isi.edu/nsnam/ns/ ".