Research Journal of Computer and Information Technology Sciences ISSN 2320 – 6527
Vol. 5(3), 1-6, May (2017) Res. J. Computer and IT Sci.

Modified random early detection algorithm to enhance the performance of bursty network traffic

Anup Kumar Sharma* **and Ashok Kumar Behera**
Department of CSE, Bhilai Institute of Technology, Durg, India
anupkumarr11@gmail.com

Available online at: www.isca.in
Received 10th April 2017, revised 16th May 2017, accepted 19th May 2017

Abstract

Reduction in the delay of the network traffic is an important problem in the communication network. But, it becomes hard to bring down the delay with boosting the performance. The reason for this, to minimize the delay there must be a small queue but to reduce the packet drop there must have a large queue. In our proposed modified RED, the Explicit Congestion notification marked and network traffic packets will be dropped when only excess congestion occurs at router in TCP/IP network. The main aim is to provide better congestion control over the bursty traffic with maintaining the advantages of RED. Due to exponential increases in network traffic, to minimize the increasing packet loss rates, the IETF (Internet Engineering Task Force) consider the establishment of AQM techniques such as RED, REM. We use ns2 to simulate the modified RED and demonstrate that the modified RED gives better delay reduction and higher link utilization in case of bursty network traffic.

Keywords: Random early detection (RED), Congestion control, Active queue management (AQM), Bursty traffic, Drop tail.

Introduction

The typical queue management technique for router, drop tail and random drop were used with the TCP, that fixes the buffer size as maximum buffer size for every router. Router accepts the packet till the buffer becomes full, afterwards it starts dropping of newly arriving packet. This mechanism is termed as Drop Tail. This mechanism has various disadvantages. The two-main important disadvantages[1]. Firstly, Lock-Out Problem; In this it might be possible that, drop tail grants one or some more flows to occupy the full queue space. Because of this it stops alternative flows from getting space in the queue. Other drawback is full-queue problem; this problem occurs when the queue becomes full or it is very nearly to full. This results in delay for packet delivery from source to destination.

Many AQM techniques has been proposed in past years[2]. The Random early detection algorithm makes use of average queue size to tackle with the problem in traditional mechanisms for congestion avoidance and control[3]. Adaptive virtual queue (AVQ) is another mechanism which introduced a virtual queue length whose capacity is less than actual link capacity[4].Another AQM mechanism is Random exponential marking(REM),which is very similar to RED mechanism except the way congestion is measured and with a different marking probability function[5].

REM tries to attain negligible delay and high link utilization by decoupling performance measure from congestion measure. Aweya J. et al. suggested a method to improve the effectiveness of RED mechanism by dynamically altering the threshold settings when the number of connections changes[6,7].

To resolve the problem, like full queue problem, with traditional queue management mechanisms, we have active queue management techniques[8]. These are proactive approach, means they start dropping the packets before buffer limit is reached, because of this, sender is notified before the queue becomes full and may respond much earlier about the congestion. Lock out problem can be resolved by using other queue management technique like random drop, because the packets are dropped from various flows rather than a single flow in these mechanisms. In this paper, we have designed a modified technique for packet drop with the help of RED algorithm and have compared with traditional packet drop technique in bursty network traffic.

Motivation for modified red

RED has indeed many enhancements over conventional mechanism but still it has some limitations regarding stability, scalability, complexity in parameter setting and ease of deployment. Many researchers proposed various methods to overcome these issues. Random early detection gateway drops the packet in two distinct ways: In first way, packet is dropped when there arises buffer overflow secondly, packet is dropped or marked when the queue size surpasses the max threshold or random nature of dropping probability[9]. So, for improvement in this mechanism, we must have to identify these dissimilar ways of packet dropping. In first way, we need to drop the packet without intimating directly to the sender regarding the occurrence of congestion. Second way in RED drop, packets may be marked instead of dropping or can be transmitted. The host can regulate his window size correspondingly.

Sender gets notified about the occurrence of congestion by the timeout or reception of duplicate ack of the sent packet. A mechanism for controlling the congestion is said the be good if it slow down the network delay, which can be obtained by retaining the avg buffer size small[10]. Most of the simulation results show that buffer overflow is the most significant cause for packet drop at the router in RED, particularly in the case of high traffic. A very important question in congestion avoidance mechanism is the approach being applied to find out the packet dropping probability. In RED algorithm, average buffer size is opted in calculation the packet dropping probability. But the problem in RED mechanism is that average buffer size changes very often because of dynamic traffic. In this condition, the packets are being dropped unnecessarily just because of overflow. This situation may result in timeout between the two parties which lowers the performance. The entire above problem with RED encourages us to develop an enhanced mechanism which doesn't forces to drop the packet because of overflow. Also, it is required from the enhanced mechanism to keep the useful characteristics of random early detection algorithm.

Modified random early detection algorithm

Generally, we observe some problems in AQM techniques. The two-main problem can be described as follows: Firstly, the AQM algorithm treats equally, traffic from all the different flows but in actual the internet traffic remainsmomentary[11]. Also, congestion window of short lived traffic is small. Therefore, in short lived flow dropping of packet not reduces congestion that much.

Secondly, Congestion is notified to sender by dropping the packet or by marking explicit congestion notification bit of header in the IP datagram. If marked datagram is considered for prior drop, transmitter find out congestion a little late which slow down the transmitter to put its window to suitable rate.

RED algorithm is the widely deployed and most popular AQM technique. RED algorithm is the techniques used for congestion avoidance at the gateway. It also detects the congestion and notifies the sender about the congestion. The random early detection gateway has been implemented to monitor avg buffer size for every queue, and selects arbitrary connection to notify about the congestion.

RED algorithm consists of following important parameter: i. Average queue size: It gives the average size of the queue. It is measured in packets or bytes and denoted as avg_L. ii. Maximum Threshold: It is the upper limit of the buffer after which all the incoming packet will be discarded and denoted as $max_{th.}$ iii. Minimum threshold (min_{th}): min_{th} is another limit of the queue after which the incoming packets are dropped or marked with a fixed probability. iv. Loss rate (P_a): P_a is the probability with which packets are discarded. v. Maximum drop probability (P_{max}): It is peak value of P_a. The packets must be discarded if the P_a value exceeds P_{max}. vi. Weights (wq): wq is represents the avg queue size.

RED algorithm can be described as follows:
RED Algorithm
Calculate average queue size avg_L
If $avg_L < min_{th}$
enqueue packets
else if $min_{th} \leq avg_L < max_{th}$
compute probability P_a
with probability P_a
Mark or drop packets from the queue
else
with probability 1-P_a
enqueue packets
else if $avg_L \geq max_{th}$
discard packet
end

To overcome these problems of RED, we choose persistent flows as an alternative of ephemeral flows for drooping of packets. Packets will be dropped only when there arises high congestion at the router in TCP/IP Network in the proposed mechanism[12].

This method is applicable with all the Active queue management techniques. An enhanced mechanism has been proposed based on the random early detection mechanism in this paper. In this random drop and front drop are also modified. Drop tail cannot be improved due to extra overhead in finding non-ECN marked or miced packets.

RED Algorithm (Modified)
For each arriving packet:
if $avg_L <= min_{th}$ **then**
enqueue packets
end
if $avg_L >= max_{th}$ **then**
Discard packets
end
if $min_{th} < avg_L < max_{th}$ **then**
if Mice == 1 **or** ECN == 1 **then**
enqueue packets
end
Drop or mark the packets with probability P_a
end

Simulation Environment

The proposed modified RED algorithm has been simulated in the simulator, NS2 and the performance has been studied and analyzed by plotting different networking parameters in the X graph plotter.

The simulation topology with one bottleneck link is shown in the Figure-1.

Table-1 gives parameters and its values used in the simulation.

Table-1: Simulation Parameter

Parameter	Value
Network Simulator Version	2.35
Duration	100 Seconds
Traffic Type	FTP
Start Time	0
End Time	100
TCP Packet Size	552 bytes
Actual TCP Packet Size (With Header)	592 bytes
Bottleneck Queue Length	100
No. of Packet Flow	3
Link Bandwidth	10 Mbps
Link Delay	1 msec
TCP Type	TCP RENO

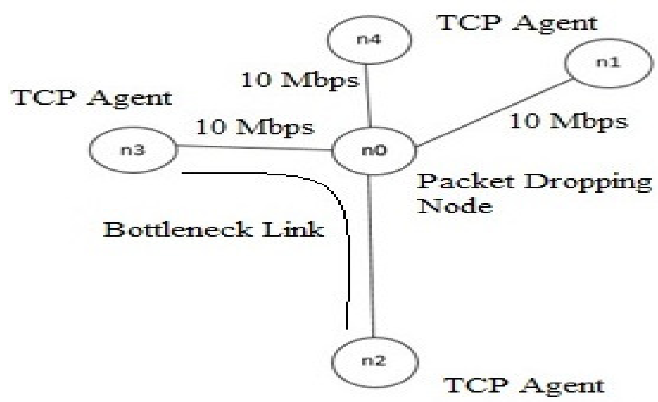

Figure-1: Topological Structure for Simulation.

Simulation result

We examine the performance of propose modified RED based on queue size i.e. average and current queue size, window size, packet dropping throughput of dropping node for bursty network traffic. We have used X graph and trace graph for plotting the graph.

Comparing the Queue Size: The queue size for both the RED and Modified RED are given in figure below. Figure-2 describe the queue size at the router and Figure-3 and Figure-4 shows the average and current queue size respectively for modified RED.

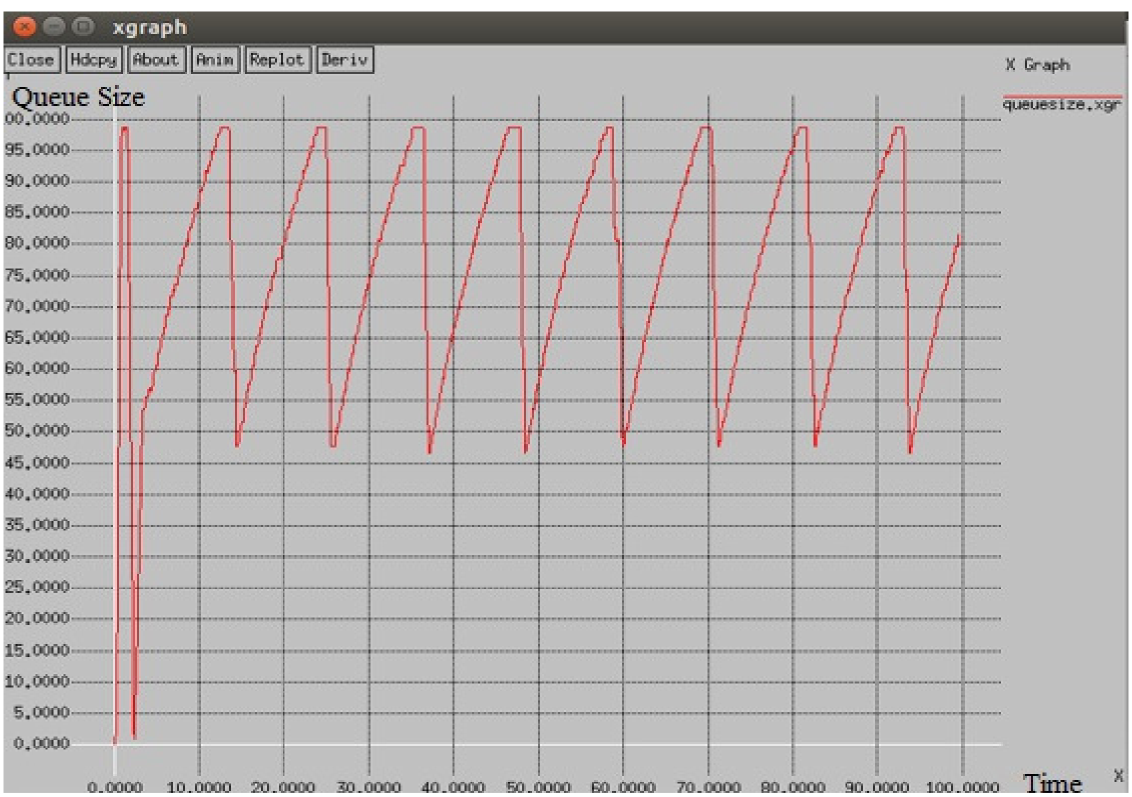

Figure-2: Queue Size for RED.

The window size of all the three flows have been analyzed for both the RED and Modified RED. We see that the window size of all the three flows is smaller than the modified RED. This indicates the better utilization of the link which results in the high performance. This shows that we are getting better performance from modified RED in case of bursty network traffic.

Packet dropping probability: In Figure-5, we describe the packet dropping probability for RED. It follows a similar pattern after some period of time. Figure-6 shows that the modified RED is not having the same periodic dropping probability pattern. Also, we get the better dropping probability in modified RED in contrast to original RED.

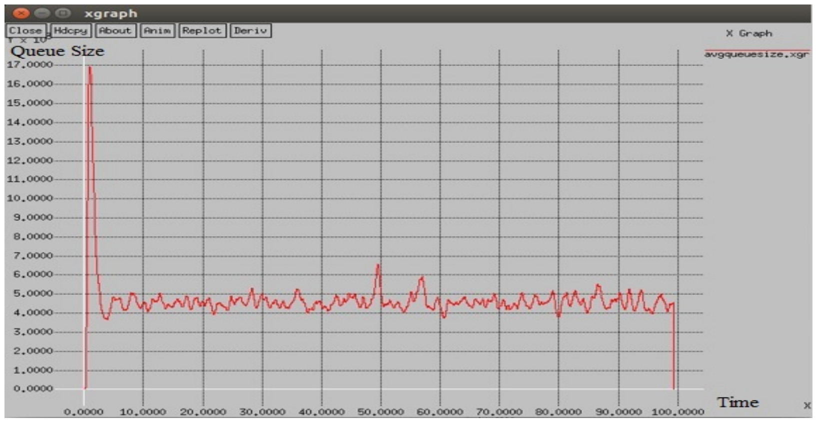

Figure-3: Average Queue Size for Modified RED.

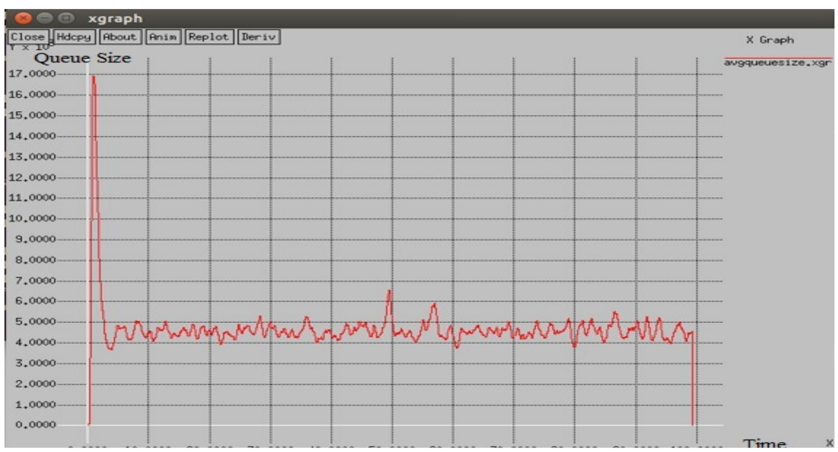

Figure-4: Current Queue Size for Modified RED.

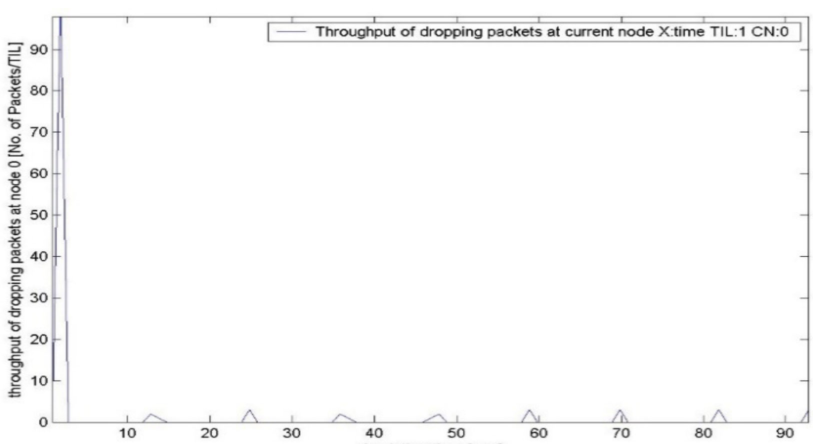

Figure-5: Packet Dropping Probability RED.

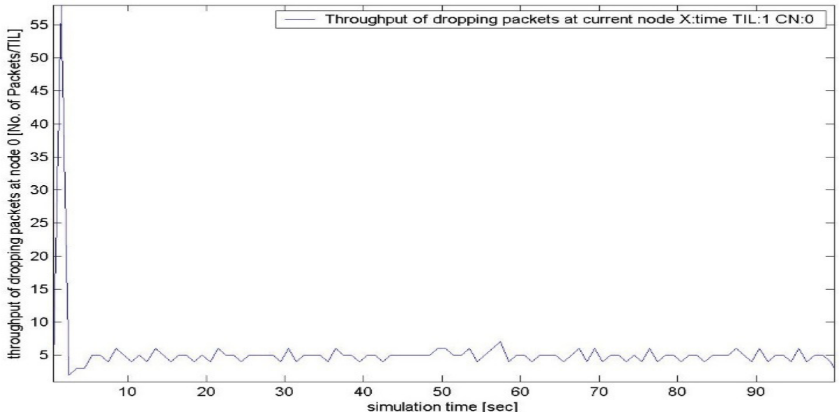

Figure-6: Packet Dropping Probability Modified RED.

Conclusion

In this paper, we have designed a modified Random Early Detection algorithm to provide a better congestion control in bursty network traffic. Designed algorithm also boosts the response time of the network traffic. This method also preserves the other advantages and the easiness of original RED, also it can be applied upon other enhanced RED techniques. Simulation result shows that we are getting an enhanced performance with modified RED in the case bursty network traffic as compared to original RED.

There are several further research areas in the queue management mechanism for various kind of traffic. The future work of the proposed modified RED algorithm is the analysis of performance and assessment of stability. Another area for research is the application of RED algorithm to a more complex network topology as well as to more discrete event description of the network. The investigation of computational complexity with respect to RED in the implementation is also extension of this work.

References

1. Braden Bob, Clark David, Crowcroft Jon, Davie Bruce, Deering Steve, Estrin Deborah, Floyd Sally, Jacobson Van, Minshall Greg, Partridge Craig, Peterson Larry, Ramakrishnan Kadangode, Shenker Scott, Wroclawski John and Zhang Lixia (1998). Recommendations on Queue Management and Congestion Avoidance in the Internet, Internet Engineering Task Force, RFC 2309.

2. Long L., Aikat J., Jeffay K. and Smith F. (2004). Differential congestion notification: taming the elephants. 12th IEEE International Conference on Network Protocols, 118-128.

3. Floyd S. and Jacobson V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4), 397-413.

4. Kunniyur S. and Srikanth R. (2001). Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. in Proc. ACM Sigcomm, San Diego, CA, 31(4), 123-134.

5. Athuraliya S., Low S.H., Li V.H. and Yin Q. (2001). REM: Active queue management. *IEEE Network*, 15(3), 48-53.

6. Aweya J., Ouellette M., Montuno D.Y. and Chapman A. (2001). An adaptive buffer management mechanism for improving TCP behavior under heavy load. Proc. IEEE Int. Conf. on Communications (ICC), 10, 3217-3223.

7. Long L., Aikat J., Jeffay K. and Smith F. (2007). The Effects of Active Queue Management and Explicit Congestion Notification on Web Performance. IEEE/ACM Transactions on Networking, 15(6), 1217-1230.

8. Floyd S. (1997). RED: Discussions of Setting Parameters. November, "http://www.icir.org/floyd/RED parameters.txt".

9. Braden Bob, Clark David, Crowcroft Jon, Davie Bruce, Deering Steve, Estrin Deborah, Floyd Sally, Jacobson Van, Minshall Greg, Partridge Craig, Peterson Larry, Ramakrishnan Kadangode, Shenker Scott, Wroclawski John and Zhang Lixia (1998). Recommendations on queue management and congestion avoidance in the Internet. RFC 2309, 1-17.

10. Dadhania M.K. and Kumar K.V. (2013). Modified RED Algorithm to Improve the Performance of Web Traffic. IEEE, 187-194. DOI 10.1109 /ACCT.2013.45

11. Feng G., Agarwal A.K., Jayaraman A. and Siew C.K. (2004). Modified RED Gateways Under Bursty Traffic. IEEE Communications Letters, 8(5), 323-325.

12. Lemeshko O., Ali S. and Simonenko O. (2015). A Queue Management Model on Router of Active Network. CADSM 2015, 419-421.

13. Baldi S., Elias B.K., Pitsillides A., Lestas M., Ioannou P.A. and Wan Y. (2016). Adaptive Optimization for Active

Queue Management supporting TCP Flows. American Control Conference (ACC). 751-756.

14. Han C., Diao Q., Chang S., Song Bi, Liu L. and Pang Z. (2016). Optimal Active Queue Management for Internet with Reduced-Order State-Observer. IEEE, 978-1-4673-9714-8, 1260-1263.