Volume 3, Issue 9, September 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Congestion Avoidence Approach in Internet Application

Shubham Asati
School of Computer Science and Engineering
VIT UNIVERSITY, Vellore, India

Ghanshyam Gagged
School of Computer Science and Engineering
VIT UNIVERSITY, Vellore, India

Krishnakant Pandey
School of Computer Science and Engineering
VIT UNIVERSITY, Vellore, India

Vairamuthu S
School of Computer Science and Engineering
VIT UNIVERSITY, Vellore, India

Abstract- Today's everyone wants good quality of services and these demands can be completed by maintaining the congestion in network. Congestion in a network may occur if the load on the network –the number of packets sent to the network-is greater than the capacity of the network- the number of packets a network can handle. So many algorithms were proposed for congestion control and congestion avoidance. This paper focuses on algorithms for congestion avoidance specially we studied RED algorithm and finally we try to omit out the bottleneck of existing algorithms by presenting a new algorithm which will eliminate the problems in current congestion avoidance algorithms, so that we increase our quality of service.

Keywords- Congestion Avoidance, RED, ECN, AQM

I. INTRODUCTION

Recent growth in the field of computing and communications has tremendously increased demand for fast and new communication services. Congestion problem has become more severe as the computer network growth is very fast now days. As Network bandwidth is shared by users, Users want to get bandwidth as much as possible. However, there's always limited bandwidth, when the bandwidth cannot satisfy every user is trying to get it, so here congestion might occur. That's why there is congestion avoidance algorithms implemented in TCP at the users' end. To avoid congestion we cannot simply rely on end host only. In the current scenario of Internet, the TCP transport protocol detects congestion only after a packet has been dropped at the gateway, however to have large queues is undesirable. Therefore, with increasingly high-speed networks, it is important to have mechanisms that keep average queue sizes low but throughput high. In the absence of proper feedback from the gateway, transport-layer protocols could induce congestion from the estimated bottleneck service time or from changes in throughput or end-to-end delay, the possibilities of routing changes or other difficulties in distinguishing propagation delay from persistent queuing delay. Here we are mainly concerned with congestion avoidance rather than congestion control. There is very subtle difference between them which is not so easy to understand. Briefly we can say that congestion avoidance algorithm allows the network to operate in the region where there is low delay and high throughput. These algorithms prevent the network from congestion and packet lost.

There are so many algorithm given for congestion avoidance, mainly we are concerned with active queue management architecture. Active queue management based algorithm is a new type of algorithm which controls the congestion before the congestion occurs. It runs on gateways, and sends explicit and implicit feedback to the end nodes. Due to this architecture many classical congestion algorithm which are suffering from many limitation are removed i.e. global synchronization and bias against busty flows problem has been removed by reducing the discard rate. For maintaining the queue length active queue management algorithm will drop or mark the packet. For controlling the congestion usually the packet is dropped, so to prevent this high loss we will deploy ECN for congestion notification. By ECN, Gateways can mark the packet instead of discarding them, so that source node can take proper action by slowing down their data rate. In this paper the performance of existing active queue management algorithm i.e.

RED and proposed algorithm is compared with the help of RTT value and ECN support under heavy traffic. With the help of ECN we will show that our proposed algorithm is superior to existing algorithm. This paper is organized in four section, first section describe about the background history of congestion algorithm ,Second section describes about our newly proposed algorithm ,third section compare the performances of existing algorithm versus proposed algorithm and final section concludes the paper work.

© 2013, IJARCSSE All Rights Reserved

II. LITERATURE SURVEY

Various researchers analyze various congestion avoidance algorithms, starting with Drop Tail [1] which is very simple and there is no need of state information. But the main disadvantage is that this algorithm is lacking in quality of service, no fairness and majorly global synchronization. DECbit [2] is another algorithm which is simple, distributed, optimized and low overhead also provides good fairness, but it uses simple averaging and it is also biased against busty traffic. RED [3]is also very famous active queue management algorithm which is unbiased for various busty traffic but this algorithm is very sensitive to parameter setting. There are so many variation of RED algorithm i.e. Gentle RED [4], Adaptive RED[5], Flow RED[6] and RED with preferential dropping algorithm[7]. Proportional Integral Controller Algorithm [8] is also an active queue management algorithm which is fast and robust and has less queue oscillation but the main problem in this algorithm is estimation and setting of constants. CHOKE algorithm [9] is simple, stateless and easy to implement but main issue is fairness and scalability problems.REM (Random Exponential Marking)[10] is a low packet loss, high link utilization , scalable and low delay algorithm but based on global parameter and it also lacks in QOS. FQA (Fair Queuing Algorithm) [11] is a bound to delay algorithm but very expensive to implement .SFQA (Stateless Fair Queuing Algorithm) is an extension of above previous algorithm which reduced the look up cost and it provides fairness but consequently it is complicated and there are more incomplete queues. CSFQA (Core Stateless Fair Queuing Algorithm)[12] which is fairness algorithm but limitation is that it contains packet header as extra field. VQ (Virtual Queue) [13] Algorithm is the high link utilization algorithm but has fixed type of virtual queue.

III. PROPOSED WORK

As above we discussed RED Algorithm which is simple, fair active queue management (AQM) algorithm but the main limitation of this algorithm is sensitivity towards parameter setting. So here we propose new algorithm which is fundamentally a different AQM algorithm based on link utilization history and uses packet loss to manage congestion. In this algorithm we will use a single probability to mark packet when they are in queue we will increment the probability marking when router buffer is over flow and a packet is dropping continuously. So by increasing the rate it sends congestion notification similarly if router buffer is empty then algorithm decrement the probability marking. With the help of simulation and experimentation we will show the advantage of this algorithm over the existing algorithm.

Table1: Proposed Algorithm for Congestion Avoidance

Upon Packet loss (Or Queue> L) event:
If((Now – last update)>freeze _time) then
$P_{an} = P_n + d_i$;
Last update = now

Upon link Idle Event:
If((Now – last_update)> freeze_time)then
$P_n = P_n - d_j$;
Last_update = now

The algorithm which we are proposing in this paper assumes that congestion level doesn't directly reveal with queue length. Hence the packet marking probability doesn't get updated with the queue length. Instead of that proposed algorithm update the packet marking probability (Pn) by using queue overflow and idle event history.

The parameter used by the algorithm: The first two parameter analysis the amount by which P_n is incremented in the case of the queue overflow (d_i) or by which Pn is decremented when the link is idle (d_j) the last parameter is the minimum time interval between two successive update (Freeze_time).

IV. PERFORMANCE COMPARISONS

In this section we will compare the performance of the RED algorithm with our proposed algorithm with the help of simulation and experimentation facts. Now consider the different cases for the performance analysis.

4.1 Case 1

When we consider the throughput our proposed algorithm converges to 100% link utilization using small queue whereas RED can reach that efficiency only for large queue.

© 2013, IJARCSSE All Rights Reserved

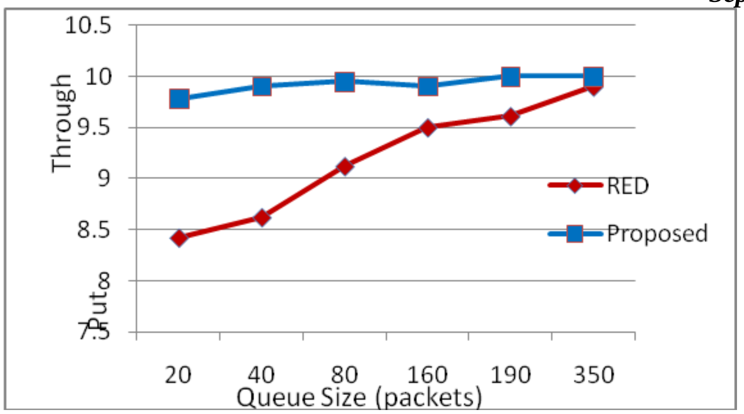

Fig 1: Total throughput over link for both RED vs. proposed

Our algorithm can successfully control the queue once the traffic behavior is learned. Whenever new source are activated, our algorithm suffer when traffic load changes. But as soon as traffic is established, our algorithm can successfully control the queue as it well know how to mark the packet. This control mechanism maintains the link at high throughput condition.

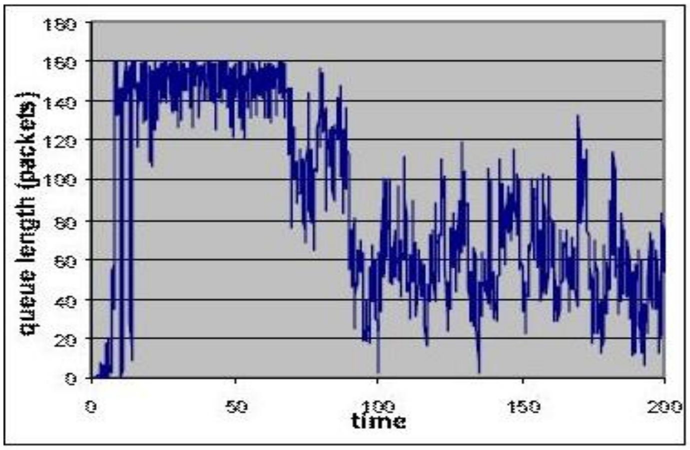

Fig 2: Queue size for RED when queue_max = 160 (large RTT)

4.2 Case 2

Compare the performance of RED algorithm and our proposed algorithm based on Drop rate technique. In low link utilization RED algorithm causes lager drop rate when the queue instability occurs which is mainly cause by queue overflow where figure as shown below. Our proposed algorithm approaches to zero drop rate when the queue size is increased. Since overflow not occurs in queue and ECN technique is used to mark the packet, so that packet loss can be kept at the minimum rate.

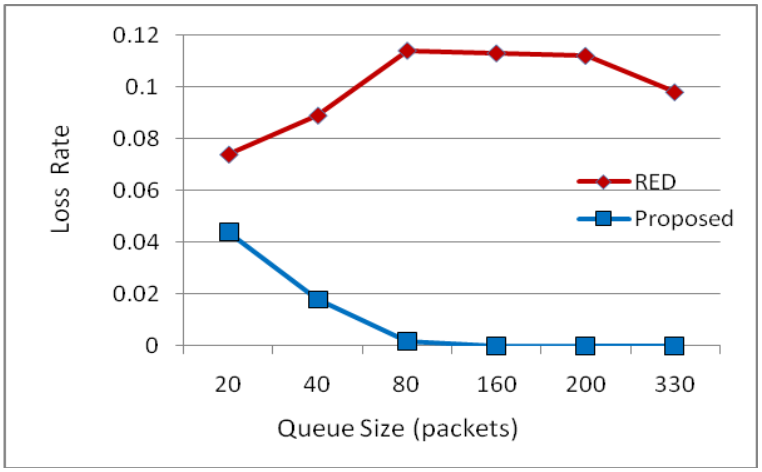

Fig 3: Total drop rate over link for both RED vs. Proposed (large RTT)

© 2013, IJARCSSE All Rights Reserved

4.3 Case 3

Compare the performance of RED algorithm with our proposed algorithm based on small RTT flows with ECN support. Below figure shows that the performance is better of the RED Algorithm under this configuration when the throughput is concerned, the reason for such result is that the ability of RED Algorithm is to control the queue successfully.

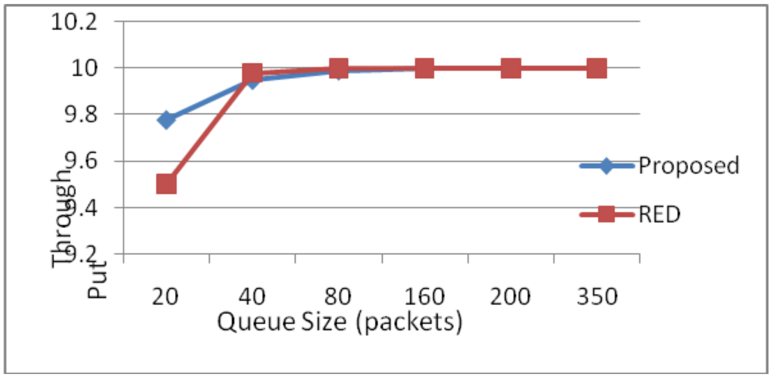

Fig 4: Total throughput over bottleneck link for both RED Vs. proposed Algorithm (small RTT)

From the fig.5 the behaviors of the RED algorithm is somewhat similar to drop tail queue and causes to much force to drop packet under this configuration.

Fig 5 shows the queue cannot control by RED algorithm and also causes high number of drop by the queue overflow which in turn results in a loss rate in shown in fig 6. Whereas our proposed algorithm can still successfully control the queue, that can keep the link at low loss rate.

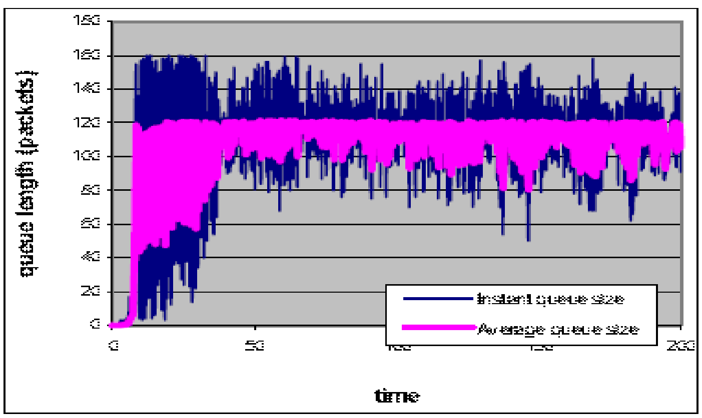

Fig 5: RED average and instant queue size graph for q_max = 160 packets (small RTT)

The important observation in this cases is that the queue cannot be controlled by the RED algorithm even it mark packet at it maximum rate that is 1. This is because of the marking rate calculation for the algorithm of RED, which uses queue length as the congestion measure. Whereas our proposed algorithm can control the queue successfully as proposed modified marking rate calculation algorithm.

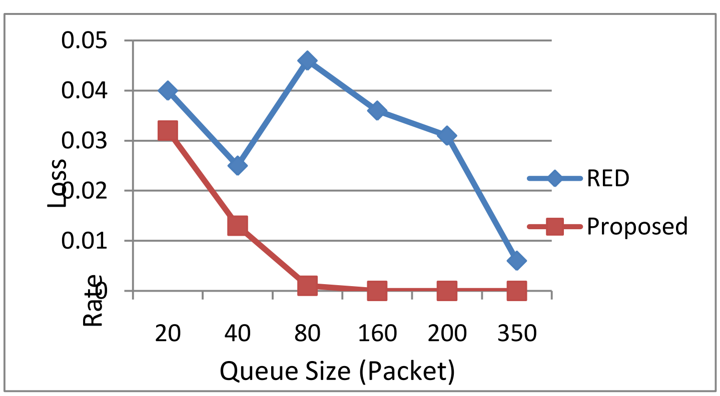

Fig 6: Total packet loss rate over link for both RED and BLUE (small RTT)

V. CONCLUSION

In this paper we discussed the limitation of RED algorithm based on active queue management and show the superiority of our proposed algorithm over the existing algorithm with the help of many parameters. Active queue management algorithm finds the congestion before the congestion occur and send notification to the source node by marking the packets due to this mechanism loss rate is decreased and throughput is increased RED is famous algorithm based on active queue management and many advantages but beside that there are many limitation, we add a new technique ECN with RED algorithm and proposed the new algorithm. Finally we proved that our algorithm is better than RED algorithm by changing the RTT value and ECN deployment in the heavy traffic. With the help of ECN the packet loss is controlled. Whereas in today's scenario ECN is not deployed generally, that's why still RED is better in some situation where ECN is not deployed with our algorithm.

References

1. E. Crawley, R. Nair, et al. A Framework for QoS-based Routing in the Internet, Network Working Group, RFC2386, Aug 1998.
2. K.K. Ramakrishnan and Raj Jain. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks, ACM Transactions on Computer Systems, May 1990.
3. B. Braden, D. Clark, et al. Recommendations on Queue Management and Congestion Avoidance in the Internet, Network Working Group, RFC2309, Apr 1998.
4. V. Firoiu and M. Borden. A Study of Active Queue Management for Congestion Control, Proceedings of IEEE INFOCOM'2000, March 2000.
5. W. Feng, D. Kandulur, D. Saha, and K. Shin. A Self Configuring RED Gateway, Proceedings of IEEE INFOCOM, Mar 1999.
6. D. Lin and R. Morris. Dynamics of Random Early Detection Proceedings of SIGCOMM, Sep 1997.
7. Ratul Mahajan and Sally Floyd. Controlling High Bandwidth Flows at the Congested Router. ICSI Tech Report TR-01-001, Apr 2001.
8. C. Hollot, V. Misra, D. Towsley, and W. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. IEEE INFOCOM, 2001.
9. Rong Pan, Balaji Prabhakar, and Konstantinos Psounis. CHOKe, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. IEEE INFOCOM, Mar 2000.
10. Steven H. Low Sanjeewa Athuraliya, Victor H. Li and Qinghe Yin. REM: Active Queue Management. IEEE Network, 2001.
11. Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queuing algorithm,SIGCOMM Symposium on Communications Architectures and Protocols, pages 1–12, Sep 1989. Austin, Texas.
12. Ion Stoica, Scott Shenker, and Hui Zhang. Core Stateless Fair Queuing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks. ACM SIGCOMM, Sep 1998.
13. R.J. Gibben and F.P. Kelly. Resource pricing and evolution of congestion control, Automatica, 1999.

© 2013, IJARCSSE All Rights Reserved