FloadAutoRED: an AQM scheme to Increase the Overall Performance in Internet Routers

Chitra Kandaswamy[1], Padmavathi Ganapathi[2]

**[1] Department of Computer Science, D.J.Academy for Managerial Excellence
Coimbatore, Tamil Nadu, India**

**[2] Department of Computer Science, Avinashilingam University for Women
Coimbatore, Tamil Nadu, India**

Abstract

The Routers in Internet face the problem of congestion due to the increased use of Internet. Active Queue Management has provided a solution to the problem of congestion control in the Internet routers. But as data traffic is bursty in routers, burstiness must be handled without comprising the high link utilization and low queuing delay. Congested link leads to many problems such as large delay, unfairness among flows, underutilization of the link and packet drops in burst. In the last decade various AQM algorithms have been evolved to solve these problems of congestion in routers. RED based algorithms use queue length as congestion indicator while some of them use flow information for more accurate congestion indication. Some AQMs have been designed to use the input rate as the congestion indicator. In this paper, we propose an AQM scheme that considers the advantages of these queue length based, rate based and flow based algorithms and satisfies the QOS requirements of the network. This proposed scheme aims to increase the overall performance even under heavy load and shields the responsive flows form unresponsive flows to offer a good Quality of Service to all users.

Keywords: *Congestion, Drop Probability, Input rate, Queue length, Stability.*

1. Introduction

A router in the Internet may receive thousands of flow at any given time due to heavy use of Real time applications. The load tends to fluctuate in an Internet router resulting in a queue oscillation. In a bursty traffic, every flow should receive its fair share while sharing the queue in the router. So a Router must be able to handle the above problems. The buffer in the routers is to be used effectively by using an efficient Active Queue Management Mechanisms. Active Queue Management tries to prevent congestion and provides quality of service to all users. A router implementing RED AQM [4] maintains a single queue to be shared by all flows that drops an arriving packet at random during periods of congestion. RED suffers from lockout and global synchronization problems when parameters are not tuned properly. The major disadvantage that exists in RED is parameter tuning problem. RED allows unfair bandwidth sharing when a mixture of traffic types share a link. In case a source sends too fast regardless of loss rate gains an unfair fraction of the bandwidth in RED AQM. In order to solve these problems, RED and its variant were proposed.

The RED based AQMs tried to solve the various problems existing with RED. Some of these AQMs tried to get rid of the parameter tuning problem in RED. While some of them tried to improve the performance compared to RED. The problem of unfairness was attempted by some of the AQMs. AdpativeRED [5], PD-RED [13], AutoRED with RED [14] tried to solve the parameter tuning problem. AQMs like MRED [6], DS-RED [15] were proposed to improve the performance measures of RED AQM. To improve fairness in case of different traffic types, some AQM started using flow based information as in Fair Random Early Detection (FRED) [8]. But FRED incurs extra implementation overhead because they collect information about the active flows. To overcome this, CHOKe [12] algorithm was proposed that simultaneously identifies and penalizes misbehaving flows. Both FRED and CHOKe were implemented in RED algorithm to calculate average queue size and packet drop probability with the additional flow manipulation to bring in fairness among the different traffic types available in the network that failed in RED.

RED based AQMs used Queue length as congestion indicator to detect congestion. Some of the AQM techniques besides using queue length as congestion indicator tried to use load and in some cases uses both queue length and load to detect congestion. AQMs like AVQ [7], Yellow [10] use load or input rate as their congestion metric to indicate congestion, whereas REM [1] calculates packet drop probability using both input rate and queue length. BLUE [3] uses link history and packet loss

as congestion indicator to compute the packet drop probability.

The objective of this paper is to propose an algorithm that considers the advantages of both queue based algorithm and rate based AQM with the flow based information. Because of the simplicity and easy implementation that exists in AutoREDwithRED, this paper takes this AQM as the base algorithm with certain modifications. Firstly to bring in the advantages of the queue based AQMs, this proposed AQM is implemented in the Queue based algorithm considering average queue size as the primary metric. Secondly it also considers the input rate to get more precise information about the congestion status. Specifically, we exhibit an AQM that removes the problems of Queue based AQMs and Load based AQMs like unfairness among the different flows by accounting the flow information. This algorithm is simple to implement and improves the performance of the routers. The rest of the paper is organized as follows: Section 2 explains the background study that includes the various AQM algorithms and its drawback. In Section 3, the concepts regarding the proposed algorithm are discussed. Simulations and results are discussed in section 4. Our conclusions are presented in section 5.

2. Background Study

The Internet routers face the problem of congestion from the birth of Internet. Research activities are carried out with the origin of various congestion avoidance mechanisms in Internet to improve the performance of Internet traffic. The origin of each of these mechanisms has revealed the inefficiency of each of the AQMs in certain circumstances especially in heavy traffic network. Currently there are numerous algorithms to handle this problem. So research in this field has become a continuous process in identifying the best Active Queue Management algorithm. Congestion leads to high packet loss resulting in high cost that is minimized by the various existing AQM schemes. The various existing AQMs detect congestion based on different factors and calculate the packet dropping probability. Based on the various information used, the degree of congestion varies and is reduced differently.

RED AQM was the first proposed by Floyd et al in 1993 with the objective of preventing congestion occurring with reduced packet loss. This AQM alleviates congestion by detecting incipient congestion early and delivering congestion notification to the host to reduce the transmission rates avoiding overflow from occurring. The success of RED depends to a large extent on the appropriate selection of the RED parameters. So this becomes a major drawback for the RED AQM implemented in the network link. RED AQM also leads to global synchronization and lock-out problem if parameters not properly tuned.

The queue-based AQM schemes use average queue-length or instantaneous queue length as a congestion indicator to calculate packet drop probability. The YELLOW AQM proves that the packet drop probability just does not depend only on the queue length rather can be calculated using the congestion indicator like input rate that helps in identifying the real congestion in the queue. In case of the rate-based AQM AVQ, it maintains a virtual queue whose capacity is less than the actual capacity of the link. However it is difficult to achieve faster response time and high link utilisation using a constant value γ. In improving the method for setting the value for γ, SAVQ [9] is proposed.

In REM, both queue length and load is used as congestion indicators. The BLUE algorithm resolves some of the problems of RED by employing two factors: packet loss from queue congestion and link utilization. So BLUE performs queue management based on packet loss and link utilization. It maintains a single probability p_m to mark or drop packets. If the buffer overflows, BLUE increases p_m to increase the congestion notification and is decreased to reduce the congestion notification rate in case of buffer emptiness. This scheme uses link history to control the congestion.

SRED in [11] pre-emptively discards packets with a load-dependent probability when a buffer in a router is congested. It stabilizes its buffer occupancy at a level independent of the number of the active connections. SRED does this by estimating the number of active connections. It obtains the estimate without collecting or analysing state information. SRED keeps the buffer occupancy close to a specific target and away from overflow or underflow. In SRED the buffer occupancy is independent of the number of connections while in RED the buffer occupancy increases with the number of connections. The hit mechanism is used to identify misbehaving flows without keeping per-flow state. Stabilised RED overcomes the scalability problem but suffers from low throughput. GREEN [2] algorithm uses flow parameters and the knowledge of TCP end-host behavior to intelligently mark packets to prevent queue build up, and prevent congestion from occurring. An improvement of this algorithm is that there are no parameters that need to be tuned to achieve optimal performance in a given scenario.

3. Proposed Algorithm

The proposed algorithm is motivated to maintain the stability of the system by adapting itself. Adapting too fast might make the system respond well to changing network conditions, but it leads to large oscillatory behavior or in the worst case even instability. Adapting it too slowly leads to a sluggish behavior and results to more losses than desired which leads to a lower throughput. In this proposed algorithm, average queue size is used as a primary metric and a virtual-queue AQM scheme is used to consider the load factor as the secondary metric. Thus the proposed algorithm tries to achieve a stable operating point for the queue size and improve the overall performance allocation irrespective of the dynamic traffic and congestion characteristics of the n flows.

As mentioned in the Introduction, the proposed algorithm - FLoadAutoRED enforces the concept of queue-based, input rate and the flow information. We propose an algorithm that modifies the AutoREDwithRED algorithm to remove its drawback and strengthen the existing features. This algorithm also indicates two thresholds on the buffer, a minimum threshold min_{th} and a maximum threshold max_{th}. In this proposed AQM, max_{th} of the queue size is not constant and it varies depending on the congestion in a traffic. Based on the level of congestion of traffic in a network at a particular instant time, max_{th} of the queue is either incremented or decremented. As the probability of packet drop depends also on max_{th}, the value of max_{th} is kept dynamically varying based on the congestion level.

This algorithm calculates the average queue size of the buffer for every packet arrival. As in Fig. 1, the input rate of the packet arrivals is tracked and the virtual queue capacity is updated. Then the load factor is calculated as in the pseudocode of the proposed algorithm. If average queue size is less than min_{th}, every arriving packet is queued. If average queue size is greater than max_{th}, every arriving packet is dropped. This results in queue size below max_{th}. When the average queue size is greater than min_{th}, every arriving packet is compared with a randomly selected packet from the queue for their flow id. If they have the same flow id, both are dropped otherwise if average queue size is less than max_{th} then the packet is dropped with the probability otherwise the new packet is dropped.

The packet drop probability P_a is calculated as in [14].

$$p_b = (Q_{ave} - min_{th}) / (max_{th} - min_{th})$$
$$p_a = p_b / (1 - count \cdot p_b)$$

Based on the load factor, the packet drop probability P_a is decreased or increased. If the load factor z is greater than $1+\delta$, then P_a is increased by (z*Delta/C) otherwise if z is lesser than 1 then the packet drop probability P_a is decreased by (Delta/z*C) otherwise no change in the packet drop probability P_a. The packet drop probability corresponds to the desired link utilization interval $[1, 1+\delta]$, the range of the desired link utilization. The packet drop probability is increased if the current link utilization is higher than the desired range, otherwise packet drop probability will be decreased if lesser than the desired utilization range.

```
Initially I_maxth = Cur_maxth = max_th
For every packet arrival {
Calculate P_t
Calculate w_q
If (P_t < 0.550) && (Q_t < I_maxth) && (Cur_maxth > I_maxth)
                Reinitialise Cur_maxth to I_maxth
Else if (P_t > 0.550) && (P_t < 0.880) && (Q_t < I_maxthresh) &&
(Cur_maxthresh > I_maxthresh)
        Increment Cur_maxth

Calculate Q_ave = Q_ave * w_q + Q_t * (1 - w_q)

VQC = γ (λ(t) – C) /* Update Virtual Queue Capacity */

z = λ(t) / VQC      /* Calculate load factor z      */

if (Q_ave < min_th)
    Forward the new packet
Else
  Select randomly a packet from the queue for their flow id
  Compare arriving packet with a randomly selected packet.
   if they have the same flow id
        Drop both the packets
   Else
     if (Q_ave > max_th)
        Drop the new packet
     Else
        Calculate the dropping probability p_a
        if (z >= 1 + δ)
                P = P_a + (z*Delta/C)
        if ( z < 1)
                P = P_a – (Delta/z*C)
        else
                P = P_a
        Drop the packet with probability P
}
```

Fig. 1 Pseudocode of FLoadAutoRED algorithm

Variables:

Q_{ave} : average queue size
p_a : current packet-marking probability
q : current queue size
p_b : temporary marking or dropping probability
w_q : queue weight
max_p : maximum dropping probability
max_{th} : maximum threshold for queue
z : Load Factor

Fixed parameters:

min_{th} :minimum threshold for queue
Delta : Reflects the response ability for the congestion.

Figure 1a Parameters of FLoadAutoRED algorithm

4. Simulations

In this section, we will use the packet-simulator ns-2 to simulate the FLoadAutoRED algorithm. In this simulation we consider a single link of capacity 1Mbps as in Fig. 2 that drops packet according to the AQM algorithm. The congestion link is in between the two routers R1 and R2. The link is shared by n TCP flows and n UDP flows. End hosts are connected to the routers using a 10Mbps link. All links have a small propagation delay of 1ms so that the delay introduced is by the buffer delay rather than the transmission delay. The maximum window size of TCP is set to 300 such that it is not a limiting factor for the flow's throughput. The TCP flows are derived from FTP sessions which transmit large size files. The UDP hosts send packets at a constant bit rate of 2 Mbps. In the simulation setup we consider 32 TCP flows and 1 UDP flow in the network. The minimum threshold min_{th} in the FLoadAutoRED scheme is set to 100 and the maximum threshold max_{th} to be twice the min_{th} and the physical queue size is fixed at 300 packets.

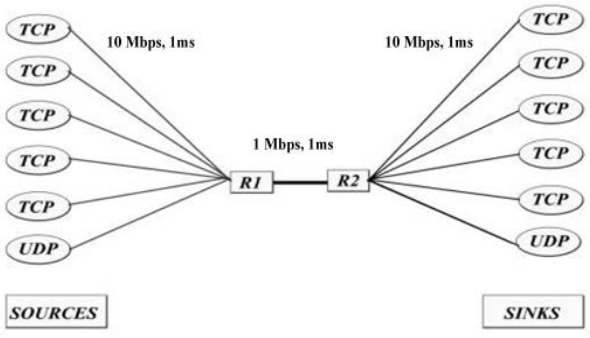

Fig. 2 Network Topology

In a dynamic varying mixture of traffic, the control parameter w_q alone does not help in achieving the stable operating point for the queue size. As shown in Fig. 3 dynamic varying parameter w_q maintains the average queue size and instability at a higher level in case of AutoREDwithRED. Though AutoREDwithRED keeps the average queue size at a stable point for the traffic consisting of adaptive flows, but for different conditions that included non-adaptive flows the average queue size projects an oscillating behaviour. But this algorithm shows a stable and a moderate average queue size compared to other AQMs as in Fig 3. The average queue size is neither too low nor high in this proposed AQM as compared to other AQMs. A very high average queue size increases the queuing delay. The average queue size should not be too low which results in poor link utilization. This proposed algorithm keeps the average queue size controlled at a moderate compared to other AQMs.

The Fig. 4 shows the adaptability of the AQM FLoadAutoRED in case of no. of TCP flows as 150. This adaptability is achieved by adapting the virtual queue capacity to the packet arrival rate. The packet drop probability is decreased or increased based on the adaptability required in terms of load factor z.

FLoadAutoRED also shows higher link utilization as in Fig. 5. Irrespective of higher link utilization it shows a lower queuing delay. The average queue size is maintained at a moderate queue size to keep the queue delay low.

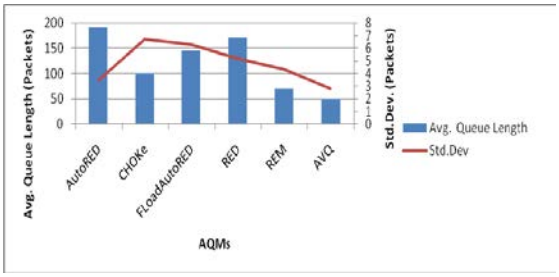

Fig. 3 Average Queue Size / Std. Dev of other AQMs with FLoadAutoRED

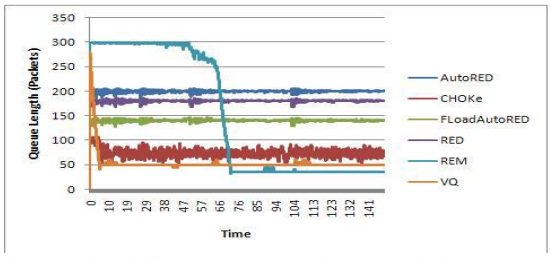

Fig. 4 Adaptability – Response Performance of other AQMs of FLoadAutoRED

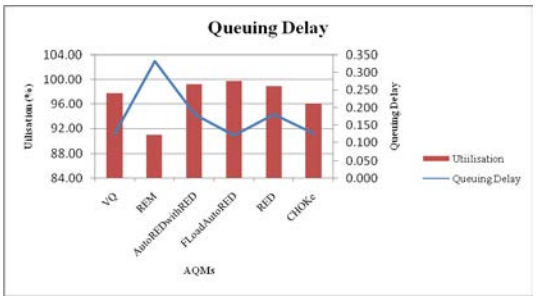

Fig. 5 Queuing Delay of other AQMs of FLoadAutoRED

5. Conclusions

This paper proposes an AQM scheme called FLoadAutoRED which aims to improve the overall performance of the Internet routers. It protects well-behaved flows from misbehaving flow and adaptive flows from non-adaptive flows. It also obtains high utilisation, low queuing delay and low packet loss with both the congestion indicators the average queue size and the input rate. This proposed AQM scheme inherits the advantages of queue length based, input rate based to bring the adaptability of the network. The performance of good service is achieved even under heavy load and protects the responsive flows form unresponsive flows to achieve a good QOS to all users by simulation.

References

[1] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: Active queue management," IEEE Network Mag., Vol. 15, pp. 48–53, 2001

[2] W. Feng, A.Kapadia , S. Thulasidasan,, "GREEN: Proactive Queue Management over a B est-Effort Network", IEEE GlobeCom, Taipei, Taiwan, November 2002

[3] W. Feng, D.D. Kandlur, D. Saha, D. Saha, "The Blue active queue management algorithms", IEEE/ACM Transactions on Networking 2002

[4] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", IEEE/ACM Trans. Networking, vol. 1, pp. 397–413, Aug. 1993.

[5] S.Floyd., R.Gummadi,S.Shenkar and ICSI, "Adaptive RED: An algorithm for Increasing the robustness of RED's active Queue Management", Berkely,CA [online] http:www.icir.org/floyd/red.html

[6] J. Koo., B. Song., Kwangsue Chung., Hyukjoon Lee., Hyunkook Kahng.,"MRED: A New Approach To Random Early Detection" 15th International Conference on Information Networking, February 2001

[7] S. Kunniyur, R.Srikant, "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management", Proceedings of ACM SIGCOMM, San Diego, 2001

[8] D. Lin, R.Morris, "Dynamics of Random Early Detection". Proceedings of ACM SIGCOMM October 1997.

[9] C. Long., B. Zhao., X. Guan, "SAVQ: Stabilized Adaptive Virtual Queue Management Algorithm", IEEE Communications Letters ., January 2005

[10] C. Long., B. Zhao., X. Guan, Jun Yang., "The Yellow active queue management algorithm", Computer Networks, November 2004

[11] T.J.Ott, T.V. Lakshman,and L.Wong, "SRED: Stablised RED", IEEE INFOCOMM, March 99

[12] R. Pan, B. Prabhakar, and K. Psounix, "CHOKe, a Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation", IEEE INFOCOMM, Feb 2000

[13] J.Sun. K. Guanrong Chen, S. Chan, M. Sukerman," PD – RED : To Improve Performance of RED", IEEE COMMUNICATIONS LETTER, August 2003

[14] S. Suthaharan, "Reduction of queue oscillation in the next generation Internet routers", Science Direct, Computer Communication, 2007

[15] B. Zheng, M. Atiquzzaman," DSRED: An Active Queue Management Scheme for Next Generation Networks" Proceedings of 25th IEEE conference on L ocal Computer Networks LCN 2000, November 2000

K.Chitra received her B.Sc. (C.Sc.) from Women Christian College, Chennai and M.Sc from Avinashilingam University for Women, Coimbatore. And she received her M.Phil degree in Computer Science from V.L.B.Janakiammal College of Arts and Science, Coimbatore in 2005. She is pursuing her Ph.D at Avinashilingam University for Women, Coimbatore. She is currently working as a A ssociate Professor in the Department of Computer Science, D.J. Academy for Managerial Excellence, Coimbatore. She has 14 years of teaching experience. Her research interests are Congestion Control in networks and Network Security.

Dr. Padmavathi Ganapathi is the Professor and Head of the Department of Computer Science, Avinashilingam University for Women, Coimbatore. She has 23 y ears of teaching experience and one y ear Industrial experience. Her areas of interest include network security and cryptography and real time communications. She has more than 80 publications at national level and International level. She is a l ife member of many professional organizations like CSI, ISTE, AACE, WSEAS, ISCA and UWA. Se is currently the Principal Investigator of 5 major projects UGC and DRDO.