# An Approach to Active Queue Management in Computer Network

Asimkiran Dandapat

*Computer Science and Engineering*
*O.P. Jindal Institute of Technology, India*
asimkiran.dandapat@opjit.edu.in,
asimkiran2004@gmail.com

***Abstract*— Active queue management is a key technique for reducing the packet drop rate in the internet. This packet dropping mechanism is used in a router to minimize congestion when the packets are dropped before queue gets full. In this paper a new framework of Active queue management namely MYRED is proposed. The goal of this new scheme is to improve the performance of AQM by keeping router queue length optimized. In RED packets are marked or dropped with a statistical probability before packet buffer overflows. In this work, a simulation based approach using NS-2 has been carried out to evaluate the performance of average queue length on simplified packet data network model.**

***Keywords*— RED, AQM, NS-2**

## I. Introduction

In the modern world Internet plays an increasingly central role, not only at the level of infrastructure, but also in culture, society and business. In internet a router maintain a set of queues, one per interface.

TCP congestion control [3][7] is widely used in the network to prevent congestion collapse. It is based on the congestion window; the sender keeps a congestion window whose size limits its transmission rate. However, these window based mechanism have a problem that TCP sender reduces its transmission rate only after detecting a packet loss caused by overflow.

Thus, queue management mechanism operating inside routers, play an important role in controlling network congestion, in conjunction with the adaptive algorithm of the TCP protocol.

Queue management algorithm manages the length of queues by dropping packets when necessary or appropriate.

An internet router typically maintains a set of queues, one per interface that hold packets maintained to go through that interface. Previously, drop-tail algorithm [2] uses queue: a packet is admitted onto the queue if the queue size is shorter than its maximum size, generally measured in packets or in bytes, and dropped otherwise.

An active queue algorithm maintains its queue by drop or mark packets before the queue gets full. TCP congestion avoidance algorithms [5][7] are used to prevent the congestion collapse over network. TCP can detect packet drops in the transmission line and treated them as indications of congestion in the network. TCP sender will take the action to these packet drops by reducing their sending rate. This reduction in sending rate translates into a decrease in the arrival rate at the router, which clear up its queue. When the arrival rate is higher than the router's dispatcher rate, the queue size will gradually increase and queue becomes full at one stage and overflow occurs.

## II. Related Work

Here is a brief overview of the active queue management algorithm RED [3] which will be used for the performance comparison.

### A. RED (Random Early Detection)

The main objective of RED algorithm is to minimize packet loss, queuing delay [3][7][6], avert global synchronization of sources, sustain high link utilization, and remove impulse against bursty sources. The basic idea behind RED[3] queue management is to detect incipient congestion early and to convey congestion notification to the end-hosts, allowing them to reduce their transmission rates before the queues in the network overflow and packets are dropped. The algorithm RED manages packet buffer by dropping or marking the packets with a probability before the buffer gets full. RED detects congestion and measures the traffic load in the network by using the average queue length *(avg)* and it is calculated using an exponentially weighted moving average filter and expressed as the following equation,

$$avg = (1 - w_q) \bullet avg + w_q \bullet q \qquad (1)$$

where $w_q$ is queue weight.

When the average queue length is less than the minimum threshold value (*minth)*, no packets are dropped. Once the average queue length exceeds the maximum threshold value, the router considers the network in congestion and drops the probability that a packet arriving at the queue is dropped depends on the average queue length, the time elapsed since the last packet was dropped, and the maximum drop probability parameter *maxp*. The drop probability *Pa* is computed as

$$P_a = \frac{P_b}{1 - cnt \bullet P_b} \qquad (2)$$

where $P_b = \max_p \bullet \frac{avg - \min_{th}}{\max_{th} - \min_{th}}$, $max_p$ is maximum value for $P_b$ and cnt is a variable that keeps track of the number of packets that have been forwarded since the last drop. If the average queue length is greater than the maximum threshold value (*maxth)*, all the arriving packets will be dropped. The Fig. 1 illustrates the working of RED algorithm.

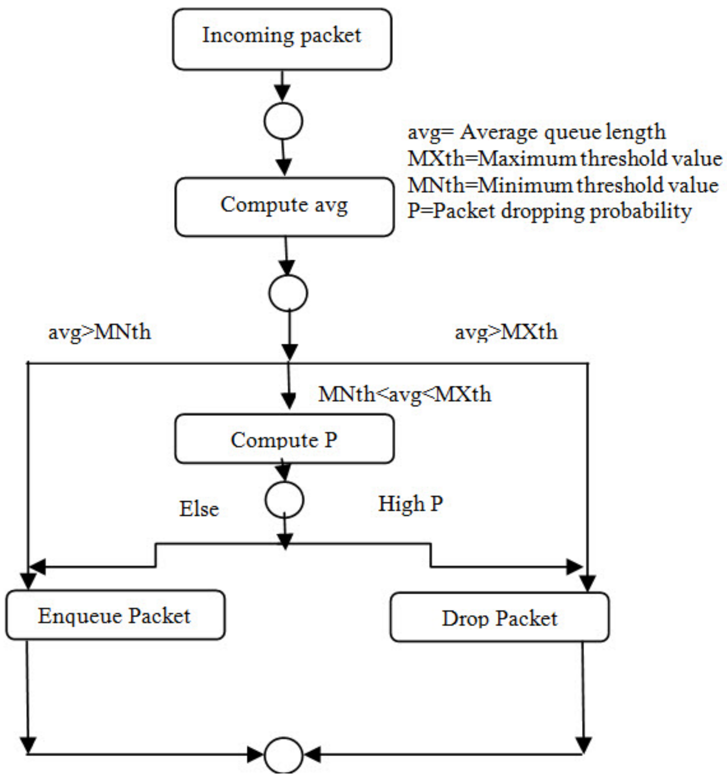


Fig. 1 Random Early Detection

## B. BLUE

BLUE [8] is another active queue management algorithm which maintains congestion control in computer network by using the rate of packet loss and link utilization history instead of queue occupancy. It operates by randomly dropping or marking the packet with explicit congestion notification mark before the transmit buffer of the network interface controller overflows. The algorithm maintains the queue by using a single probability $P_m$ to mark or drop the packets. If the queue is continually dropping packets due to buffer overflow, it increments $P_m$ and thus incrementing the rate at which it sends back congestion notification or dropping packets. Conversely, if the queue becomes empty or if the link is idle, the algorithm decreases its marking probability. The amount by which $P_m$ is increased when the queue overflows is determined by d1, while d2 determines the amount by which $P_m$ is decreased when the link is idle. Freeze time determines the minimum time interval between two successive updates of $P_m$. The Fig. 2 illustrates the working of BLUE algorithm.

Upon packet loss event:

if ( (now - last update) > freeze time ) then
$P_m = P_m + d1$
last update = now

Upon link idle event:

if ( (now - last update) > freeze time ) then
$P_m = P_m - d2$
last update = now

Fig. 2 BLUE Algorithm

## III. Proposed Scheme

RED is an improvement over the traditional drop-tail [4] queue management technique. The primary idea behind RED queue management technique is to observe incipient congestion early and to convey congestion notification to the end-hosts, allowing them to cut down their rate of packet transmission before queue overflows and the packets are dropped. Whereas BLUE [8] maintains congestion control by using the packet loss rate and link utilization history instead of queue occupancy. In a busy period, the transmitting at a rate of a single link greater than the bottleneck link capacity can cause a queue to build up just as easily as a large number of sources can. So in a heavily loaded network RED algorithm may not perform as well as the other congestion control algorithms. It is difficult to stabilize the queue size as well as to parameterize a queue to give good performance under the different network scenarios and over load level of a wide range[9][10]. In order to improve the performance of congestion routers, a new framework of AQM, namely MYRED an active queue management scheme is proposed. In the case of robust flow in the place of RED AQM the new AQM MYRED is proposed. In order to stabilize the queue length, congestion levels are proposed. It is considered that if there is no congestion then it is zero level and if it is high congestion then it is 1.

### A. Simulation Methodology

In this scheme comparison has been made between RED and MYRED according to the parameters shown in the table I. The result has been obtained using NS2 simulator in LINUX environment. In the simulation process, the topology shown in Fig. 3 is used, which consists of two senders nodes(N1, N2) and one sink node(N4), connected via router N3, with one of the queue management schemes. By varying TCP flow number in each sender node, produce different levels of traffic load and thus different levels of congestion on the bottleneck link is created. N3 has a queue buffer size of 25 packets.

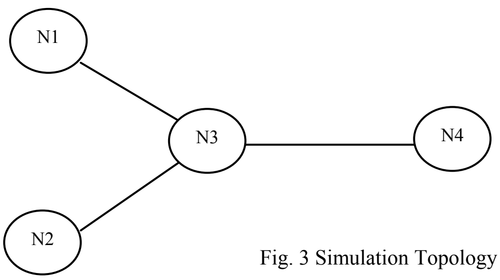


Fig. 3 Simulation Topology

TABLE I THE SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Simulator | NS2 |
| Link Bandwidth(N3-N4) | 1.5Mb |
| Packet Transmission Delay | 20ms |
| Queue Length | 25 unit(packets) |
| Congestion Level Value | 0.0,0.25,0.5,0.75 |

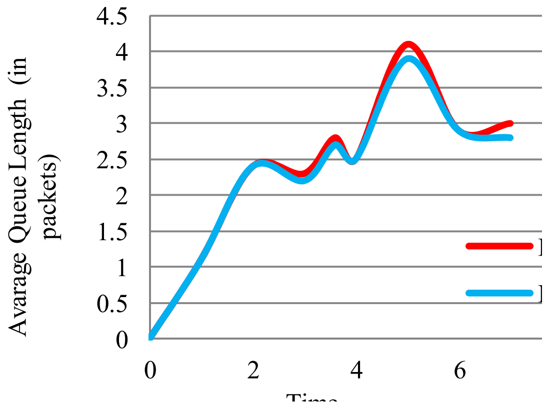


Fig. 4 Average Queue Length (in packets) w.r. to Time (in seconds)

## B. PERFORMANCE ANALYSIS

In this section, we compare the simulation results of the proposed MYRED with the existing queue management scheme RED illustrates in the Fig. 4. Though initially the performance of both schemes is similar, it has been observed here that MYRED outperform over RED with time. The parameters used for the simulations are presented in Table 1.After analysing the graph of average queue length (in packets) w.r. to time (in second) between the two active queue management algorithms RED and MYRED got the result shown in table II

I RESULT

TABLE II

| Active Queue Management Algorithm | Highest Peak of Average Queue Length (q*10^3) |
|---|---|
| RED | 4.10 |
| MYRED | 3.95 |

Here we have better performance in MYRED than RED in the respect to average queue length.

## IV. CONCLUSIONS

This proposed scheme is giving better performance to obtain the optimum routing queue length with respect to the existing active queue management algorithm RED. There are many algorithms present in routing active queue management. But here could not compare the performance of newly proposed scheme with all the available algorithms in the term of packet drop, queuing delay, end to end delay etc. Therefore there is a huge scope to work with these aspects.

## REFERENCES


[1] Babek Abbasov Mathematical and Computational Applications, “An Active Queue Management algorithm For Reducing Packet Loss Rate” Vol. 14, No. 1, pp. 65-72, 2009.

[2] Floyd, S., TCP and Explicit Congestion Notification. ACM Computer Communication Review, V. 24 N. 5, October 1994, pp. 10-23.

[3] Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance V.1 N.4, August 1993, p. 397-413

[4] P. Tinnakornsrisuphap and R. J. La, Asymptotic Behavior of Heterogeneous TCP Flows and RED Gateway, IEEE/ACM Transactions on Networking, February 2006.

[5] M. Shin, S. Chong, and I. Rhee, Dual-Resource TCP/AQM for Processing-Constrained Networks, IEEE/ACM ToN. April 2008.

[6] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the Internet”, IEEE/ACM Transaction on Networking, vol. 7, pp, 458-472, August 99.

[7] B. Braden et al, ‖Recommendations on queue management and congestion avoidance in the internet,‖ Request for Comments 2309, Network Working Group, April 1998.

[8] W. Feng, D. D. Kandlur, D. Saha, and D. G. Shin, “BLUE: A New Class of Active Queue Management Algorithms”, *Technical Report* CSE-TR-387-99, Dept. of EECS, University of Michigan, April 1999.

[9] Patel, S. “Performance analysis and modeling of congestion control algorithms based on active queue management”IEEE, 449 – 454, Dec 2013

[10] Jong-hwan Kim, Hyunsoo Yoon and Ikjun Yeom “Active queue management for flow fairness and stable queue length”, IEEE Transactions ,Vol.22 , 571 – 579, May 2010