Volume 2, No. 4, July-August 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
CASE STUDY & SURVEY REPORT
Available Online at www.ijarcs.info

QoS Congestion Control AQM Algorithms: A Survey

V. Sinthu Janita Prakash*
Head, Dept of Computer Science,
Cauvery College for Women,
Tiruchirappalli. India.
sinthujanitaprakash@yahoo.com

Dr. D. I. George Amalarethinam
Assoc. Prof & Director of MCA
Jamal Mohamed College,
Tiruchirappalli, India.
di_george@jmc.edu

Dr. E. George Dharma Prakash Raj
Asst. Prof in Computer Science,
Bharathidasan University,
Tiruchirappalli. India
georgeprakashraj@yahoo.com

Abstract: A Congestion control algorithm is one of the keys that keep any network efficient and reliable for the users. Many Algorithms were proposed in the literature over the years for the efficient control of congestion that occur in the network. Active Queue Management (AQM) is one such algorithm that provides better congestion control in the recent years. It works at the router for controlling the number of packets in the router's buffer by actively discarding an arriving packet if congestion occurs. The algorithms given in the literature give better delay performance and high throughput over different traffic conditions. In this paper an exhaustive survey is made on the AQM Algorithms that were proposed and the merits and demerits is presented
.

Keywords: Congestion control; AQM; queue based; rate based; RED

I. INTRODUCTION

Congestion in Internet occurs when the link bandwidth exceeds the capacity of available routers. This results in long delay in data delivery and wasting of resources due to lost or dropped packets. The primary role of a router is to switch packets from the input links to output links through buffer. Apart from forwarding the packets, routers are involved for controlling the congestion in the network. It is known from [1] that routing algorithms focus on two main concepts, namely, queue management and scheduling. Queue management algorithms manage the length of packet queues by dropping packets whenever necessary whereas scheduling algorithms determine which packets to be sent next. These algorithms are used primarily to manage the allocations of bandwidth among various flows.

In Internet, dropped packets serve as a critical mechanism of congestion notification to end nodes. The solution to the full queues problem is for routers to drop packets before a queue becomes full, so that end nodes can respond to congestion before buffers overflow. Such approach is called as "Active Queue Management (AQM)"[2].

II. ACTIVE QUEUE MANAGEMENT ALGORITHMS

According to the metrics used to measure congestion, AQM algorithms can be classified into three categories queue-based, rate based and a combination of both [3]. In queue-based schemes, congestion is identified by observing the average or instantaneous queue length and the control aim is to stabilize the queue length. The drawback of queue-based schemes is that a backlog is inherently necessitated. Rate-based schemes accurately predict the utilization of the link, determine congestion and take actions based on the packet arrival rate. Rate-based schemes can provide early feedback for congestion. Figure -1. Shows the categories of AQM algorithms

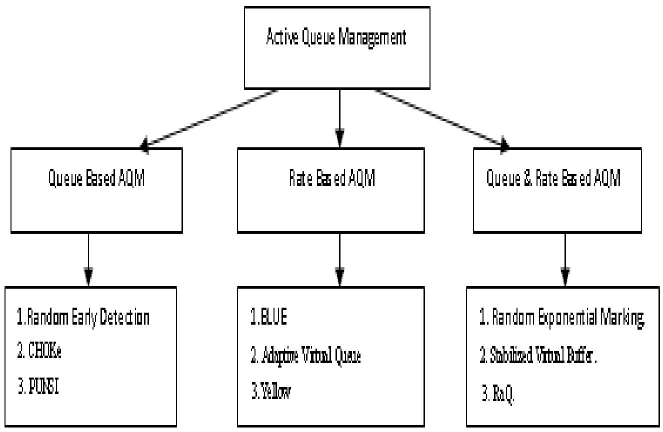

Figure – 1: Classification of AQM Algorithms

A. *Queue Length based AQM Algorithms*

In queue-based schemes, congestion is observed by average or instantaneous queue length and the control aim is to stabilize the queue length. The drawback of queue-based schemes is that a backlog is inherently necessitated. Following are some of the popular Queue Length based AQM Algorithms.

a. *Random Early Detection (RED)*

The oldest active queue management scheme called as Random Early Detection (RED) [4] lessens congestion by detecting the initial congestion early. It delivers congestion notification to the end source allowing them to reduce the transmission rates before overflow occurs. Since RED acts in anticipation of congestion, it does not suffer from the "Lockout" and "Full queue" problems present in the drop tail mechanism. The "Lock-Out" phenomenon is often due to the result of synchronization or other timing effects. In routing the packets there is a tradeoff between delay and throughput. If the queue is full or almost full, an arriving burst will cause multiple packets to be dropped. This can result in global synchronization of flows throttling back, a sustained period of lowered link utilization, reducing overall throughput. This situation is called as "Full Queue". By keeping the average queue size small, RED reduces the delays experienced by most flows. The effectiveness of RED depends to a large extent, on the appropriate selection of the parameters.

b. CHOKe

The design goal of CHOKe [5] is to keep the algorithm as simple as possible while controlling unresponsive flows. A small modification is proposed over a plain FCFS queue with RED active queue management to achieve this algorithm. When a packet arrives, if the average queue size is greater than a particular level, CHOKe draws a packet randomly from the buffer and compares it with the arriving packet. If they are from the same flow, then they are both dropped; otherwise the arriving packet is accepted in the queue with a drop probability that is computed by RED. The basic idea behind CHOKe is that a FIFO queue is more likely to have packets that belong to unresponsive flows more than those of responsive ones, and they are more likely to be chosen for comparison. Therefore, packets from unresponsive flows are likely to be dropped more often. CHOKe is very simple to implement, maintains minimum state information, and controls unresponsive flows, especially for CHOKe with multiple drop candidates.

However, CHOKe can control unresponsive flows only if there are more packets from those flows in the buffer at the time of congestion. This is due to the fact that CHOKe does not keep track of those unresponsive flows. In addition, some responsive flows might be punished unnecessarily as a result of its probabilistic algorithm.

c. Penalizing Unresponsive Flows without State Information (PUNSI)

Another algorithm, which deals Queue Length, is the PUNSI algorithm [6]. It prevents unresponsive flows from dominating available bandwidth shared with responsive flows. This is done by penalizing packets from unresponsive flows with a higher probability than those from responsive flows. It is motivated by the observation that unresponsive flows tend to generate traffic of higher rates than responsive flows and that, when a packet is dropped due to buffer overflow, fellow packets from the same flow seem to be found in the buffer among those having joined recently. This algorithm first allocates good fair share of bandwidth among all flows passing through a router and achieves this without per flow information. Queuing algorithms with good fair sharing of bandwidths and stateless information are important since they reduce the complexity due to large overhead caused by more number of flows as against algorithms like Flow Random Early Drop (FRED) which maintain per flow status. CHOKe algorithm [5] penalizes not only high bandwidth UDP flows but also TCP ones. Several packet losses in a short period worsen TCP performance significantly It doesn't work well if there are only a few packets from unresponsive flows in the queue. These two shortcomings of CHOKe are overcome by PUNSI algorithm that penalized UDP flows more effectively in accordance with its burstiness.

There are many other Queue Length Based Algorithms such as SHRED, FRED, HRED, CBT 12, ARED, SRED ,RARED, DSRED , QVARED, MRED, ADAPTIVE RED, PDRED, STOCHASTIC RED and LRE.

B. Rate Based AQM Algorithms

Rate based AQMs determine congestion and take actions based on packet arrival rate. The goals of the rate based AQMs are to reduce the rate mismatch between enqueue and dequeue, and achieve low loss, low delay and high link utilization. Since the queue lengths is a cumulative difference value of rate mismatch between enqueue and dequeue, queue merit is insensitive to current queue arrival rates. This produces the conservative and aggressive packet marking behavior when the queue length is small or large. Following are some of the popular Rate based AQM Algorithms.

a. Blue

One of the natural problems with the AQM algorithms is that they use queue length as the indicator of the severity of congestion. With this background, a fundamentally different AQM, called BLUE, [7] is proposed, implemented and evaluated. BLUE uses packet loss and link idle events to manage congestion. BLUE maintains a single probability, to mark or drop packets when they are enqueued. If the queue is continuously dropping packets due to buffer overflow, BLUE increments the single probability, thus increasing the rate at which it sends back the congestion notification. Conversely, if the queue becomes empty or if the link is idle, BLUE decreases its marking probability. This allows BLUE to know the correct rate it needs to send back congestion notification. The most important consequence of using BLUE is that congestion control can be performed with a minimal amount of buffer space. This reduces end-to-end delay over the network, which in turn improves the effectiveness of the congestion control algorithm.

b. Adaptive Virtual Queue (AVQ)

Another rate based AQM called Adaptive Virtual Queue algorithm for active Queue Management (AVQ) maintains a virtual queue whose capacity is less than the actual capacity of the link [8]. When a packet arrives in a real queue, virtual queue is also updated to reflect the new arrival. Packets in the real queue are dropped or marked when the virtual buffer overflows. The virtual capacity at each link is then adapted to ensure that the total flow entering each link achieves desired utilization of the link. When dropping is employed at the routers, the AVQ performs better than other AQM schemes in terms of utilization and average queue length but the fairness can be improved using probabilistic AQM scheme like RED on AVQ.

c. Yellow

Yellow active queue management algorithm [9] uses the value between the input rate and link capacity as the primary metric. Therefore the advantages of rate based AQMs are inherited. Also, queue size is made as a secondary metric. Queue length affects the load factor using Queue Control function, which is computed by a non-linear hyperbola function of instantaneous queue length and reference queue size. Known from other rate based schemes, Yellow provides an early controlling queuing delay maintaining the main load merit. The average queue length and Standard deviation of queue length of Yellow are little affected by the introduction of the UDP flows. May be too aggressive with the globally asymptotic stability condition which leads to under utilisation. There are other Rate Based ATM Algorithms such as SFB, SFED, FABA, SAVQ, EAVQ, LUBA, RAQM, PRC, REAQM.

C. Queue Length and Rate Based AQM Algorithms

a. Random Exponential Marking (REM)

Random Exponential Marking [10] is an active queue management Algorithm that aims to achieve both high utilization and negligible loss and delay in a simple and scalable manner. The key idea is to separate congestion measure from performance measure such as loss, queue length or delay. While congestion measure indicates excess demand for bandwidth and must track the number of users, performance measure should be stabilized around their targets independently of the number of users.

b. Stabilized Virtual Buffer (SVB)

Stabilized Virtual Buffer (SVB) [11] also similar to REM, considers both the packet arrival rate and queue size to stabilize them around target value but unlike REM, it maintains a virtual queue and responds to the traffic dynamics faster for better stability, especially in the presence of short flows. While the virtual concept queue is similar to AVQ, this SVB considers both arrival rate and queue length. Unlike AVQ where the service rate of the virtual queue is adaptable and packet is dropped or marked whenever the virtual queue overflows the physical buffer limit, in SVB the service rate is fixed as link capacity of the real queue and adapt the limit of the virtual buffer to the packet arrival rate. Another difference with respect to AVQ is that the incoming packets in SVB are then marked with a probability, which is calculated based on both the current virtual buffer limit and virtual queue occupancy.

c. RaQ

RaQ [12] uses the input rate and current queue length to calculate the packet dropping/marking probability. From the point of control theory, RaQ can be seen as dual loop feedback control. The inner loop is rate feedback and outer loop is queue length feedback control. Thus the rate feedback control enables RaQ to respond congestion quickly, so that it can decrease the packet loss due to buffer overflow, and queue length feedback control stabilizes RaQ's queue length around given target. So it can achieve predictable queuing delay and lower delay jitter.

III. SUMMARY OF AQM ALGORITHMS

There has been many AQM Algorithms that have been developed since 1997, a few of them are considered in this study. The goals of AQM are to maintain a stabilized queue, to achieve high resources utilization and lower queuing delay. The "Lock out" and "Full queue" problems of tail drop mechanism are the issues that are being considered while developing any new AQM mechanisms. Table.1 summarises the merits and demerits of various AQM algorithms discussed so far.

Table.1 Merits and demerits of various AQM algorithms.

AQM Schemes	AQM Algorithms	Merits	Demerits
Queue based	RED	Early congestion detection Keeps average queue size small; Reduces delay Global synchronization mitigated	Effectiveness depends on the selection of parameters Insensitive to traffic load and drain rate
	CHOKe	Simple to implement; Better fairness Controls Unresponsive flows	Some responsive flows may be punished Poor utilization
	PUNSI	Better TCP performance Prevents unresponsive flows from dominating	The accuracy of estimate is susceptible under web-like flows
Rate Based	BLUE	Easy to understand ; High Throughput Reduces end to end delay	No early congestion detection Slow response
	AVQ	Faster Response Better utilisation and average queue length	Queuing delay increases with increasing congestion Fairness can be improved
	YELLOW	Fast response ; Queue stability Small queuing delay Average queue length not affected by UDP flows	Jitter Maybe too aggressive with the globally asymptotic stability condition which leads to under utilisation.
Queue& Rate Based	REM	Low delay and small queues Independent of number of users	Some complexity due to parameters Low throughput for web traffic
	SVB	Responds to traffic dynamics faster Better stability Better goodput and less loss rate	Implementation is complicated Requires a proper intelligent marking mechanism at the edge routers
	RaQ	Respond to congestion quickly Decrease in packet loss due to buffer overflow Lower delay jitter	Implementation is complicated Queuing delay increases with increasing congestion

IV. CONCLUSION

In this paper, we have presented a survey on advances in the area of active queue management. Also, we classified the algorithms according to the type of metrics they used as congestion measure. There are more number of algorithms in Queue Length Based algorithms such as SHRED, FRED, HRED, CBT 12, ARED, SRED, RARED, DSRED , QVARED, MRED, ADAPTIVE RED, PDRED, STOCHASTIC RED and LRE and in Rate Based

Algorithms such as SFB, SFED, FABA, SAVQ, EAVQ, LUBA, RAQM, PRC, REAQM which can be considered surveyed in the future.

V. REFERENCES

[1] K. Satyanarayan Reddy, C. Lokanatha Reddy "A survey on congestion control mechanisms in high speed networks" International Journal of Computer Science and Network Security (IJCSNS) Vol. 8, No. 1, pp. 187 – 195, January 2008.

[2] V. Firoiu and M. Borden. "A study of active queue management for congestion control" in Proceedings of IEEE INFOCOM, pp.1435 – 1444, Mar. 2000.

[3] Jaesung Hong., Changhee Joo., Saewoong Bahk, "Active queue management algorithm considering queue and load states", Computer Communication, vol 30, Issue, pp.886-892, February 2007.

[4] D.Lin., R.Morris., "Dynamics of random early detection", Proceedings of ACM SIGCOMM, pp. 127-138 October 1997.

[5] Pan ,R., Parbhakar, B., and Psounis K., "CHOKe, a stateless active queue management scheme for approximating fair bandwidth allocation", Proceedings of IEEE INFOCOM Conference on Computer Communications, pp.942-951, March 2001.

[6] Tetsuji Yamaguchi.,Yutaka Takahashi., " A queue management algorithm for fair bandwidth allocation", Computer Communications , Volume 30, Issue 9, pp. 2048-2059, June 2007.

[7] Wu –Chung Feng., Kang G.Shin., Dilip D.Kandlur., Debanjan Saha ., "The BLUE active queue management algorithms", IEEE ACM Transactions on Networking, Volume 10 Issue 4, pp.513-528 , August 2002.

[8] Srisankar S. Kunniyur., R.Srikant " An adaptive virtual queue (AVQ) for active queue management"., IEEE/ACM Transactions on Networking, volume 12 Issue 2,pp.286-299,April 2004.

[9] Chengnian Long., Bin Zhao., Xinping Guan., Jun Yang., "The Yellow active queue management algorithm", Computer Networks,pp.525-550, November 2004.

[10] Athuraliya., D.E Lapsley., " Low robustness of real and virtual queue-based active queue management schemes", IEEE ACM transactions on Networking, Vol 13, Issue 1, pp.81-93, February 2005.

[11] Xidong Deng., Sungwon Yi., George Kesidis., Chita R.Das., " Stabilised Virtual Buffer (SVB)-An active queue management scheme for internet quality of service", IEEE Globecom November 2002.

[12] Jinsheng Sun., Moshe Zukerman ., " RaQ: a robust active queue management scheme based on rate and queue length", Computer Communications, Volume 30 Issue 8, PP. 1731-1741, June 2007.