Research Journal of Applied Sciences 8 (2): 116-124, 2013
ISSN: 1815-932X

A Simulation Study of Rate and Queue Based on Active Queue Management

Nabil Isaa Jafar
Department of Administration Unit, University of Technology, Baghdad, Iraq

Abstract: Now a days, e-Queues are built up everywhere where customer online-service is necessary such as in banks' e-Service, enterprises' e-Business, etc. In order to enhance Active Queue Management (AQM) and Quality of Service (QoS) algorithms are frequently employed due to its efficiency in congestion avoidance as well as the differentiated forwarding of packets. This research aims at developing a novel AQM algorithm to better QoS in terms of congestion prediction, packet loss, queuing delay and link utility, etc. Upon the traditional designs of AQM, this research establishes a new integrated AQM scheme (RQ-AQM) by employing current queue length and input rate to calculate the packet marking/dropping probability. In this way, the feedback rate control enables to respond to congestion rapidly, decreasing the packet loss from buffer overflow. Meanwhile, stabilizing the queue feedback control around a given target to achieving predictable queuing delay and lower delay jitter. Thus, the main feature of the design is to use coefficients of both proportional rate control and proportional integral queue length control. And to simplify parameter setting, the control parameters were scaled by the link capacity C to normalize the rate and by the Bandwidth-Delay Product BDP to normalize the queue length, respectively. The stability performance of RQ-AQM was tested via simulation under several conditions. The results proved that it is able to maintain the queue length around the given target. Also, the comparison results with other AQM schemes including RED, ARED, PI controller, AVQ and REM, demonstrated the superiority of RQ-AQM in low packet loss, faster convergence to target queue length and closest to the target queue length.

Key words: Active queue management, congestion control, queue-based, rate-based, simulation model, algorithm management

INTRODUCTION

Internet congestion occurs when the availability of a resource fails to meet the aggregate demand for the resource which results in long delays in data delivery, wasted resources due to lost or dropped packets and even possible congestion collapse in which all communication ceases in the entire network.

To control congestion, the buffer management played the first important role. Its primary objectives were low packet queuing delay and high link utilization. The traditional algorithm to manage buffer is First-In-First-Out (FIFO) with Tail-Drop which drops packets only if buffer overflows. This passive behavior results in long queuing delay and often causes the correlation among packet drops resulting in the well-known Transmission Control Protocol (TCP) synchronization problem (Braden *et al*., 1998). To mitigate such problem, Active Queue Management (AQM) (Braden *et al*., 1998) as a type of router mechanisms has been attached high attention by researchers recently. Comparing with FIOF Tail-Drop, AQM algorithms manage queue lengths by dropping (or marking if Explicit Congestion Notification (ECN) (Lin and Moms, 1997) is enabled) packets when congestion is building up (Braden *et al*., 1998) that is before the buffer becomes completely full. End systems can then react to such losses by reducing their packet rate hence avoiding severe congestion early and achieving better link utilization.

The key issues in designing AQM algorithms are how to measure link-congestion degree and how to determine the behavior of packet dropping or marking. And upon the existing related research, link-congestion is estimated through (average) queue length, input rate, event of buffer overflow or buffer empty or/and combination of them.

Amongst them, queue length is the most broadly used measurement. It uses the average (or instantaneous) queue length to compute the packet dropping/marking probability. The most typical example is Random Early Detection (RED) (Yin and Low, 2001) which is one of the first AQM mechanisms that have been proposed. It intends to avoid congestion by randomly discarding packets based on the average queue size. Other citable examples include its modifications (Floyd, 1994; Hollot *et al*., 2002), PI control algorithm (Bonald *et al*., 2000), Dynamic RED (Sun *et al*., 2003a, b), PD-Controller (Feng *et al*., 1999), etc. The event of buffer

overflow or buffer empty is more considered in BLUE (Paganini, 2001) to estimate link-congestion level. And the input rate is focused in GREEN (Kunniyur and Srikant, 2001) and Adaptive Virtual Queue (AVQ) (Sun *et al*., 2003a). Recently, Random Exponential Marking (REM) (Wang *et al*., 2003) and Virtual Rate Control (VRC) (Wang *et al*., 2004) are representative algorithms that jointly use queue length and input rate to determine the packet dropping/marking probability.

This study attempts to propose a more stable algorithm with faster response or convergence rate since the internet flows often vary and hard to be known exactly and slow response may lead to buffer overflow or emptiness or corresponding large queuing delay or low link utility (Sun *et al*., 2007). By employing input rate and current queue length, a new integrated Rate and Queue-based AQM (RQ-AQM) scheme is established to calculate the packet dropping/marking probability. Thus, the rate feedback control enables to respond to congestion rapidly which can decrease the packet loss due to buffer overflow while the queue length feedback control stabilizes the queue length around given target which can achieve predictable queuing delay and lower delay jitter.

RQ- AQM

Algorithm description: RQ-AQM is an integrated rate-based and queue-based AQM scheme. According to control theory, it can be seen as dual loop feedback control with the inner loop feedback to control the rate-based and the outer loop feedback to control the queue-length based. Specifically, it uses proportional rate control and proportional integral queue length control as follows:

$$p(t) = r_{kp}(r(t) - C) + q_{kp}e(t) + q_{ki}\int_0^t e(t)dt, \quad 0 \le p(t) \le 1 \tag{1}$$

Where:

- $p(t)$ = The packet dropping/marking probability
- $r(t)$ = The aggregate input rate of the queue
- C = The link capacity
- $e(t)$ = $q(t)-q_{ref}$, $q(t)$ is the instataneous queue length, q_{ref} is the target queue length
- r_{kp} = The proportional coefficient of rate control
- q_{kp} and q_{ki} = The proportional and integral coefficients of queue control, respectively

To achieve consistent queuing behavior regardless of link capacity and propagation delay and to simplify parameter setting, the control parameters could be scaled by using the link capacity C to normalize the rate and by the Bandwidth Delay Product (BDP) to normalize the queue length thus Eq. 1 becomes:

$$p(t) = r_{kp}\left(\frac{r(t)}{CF-1}\right) + q_{kp}\frac{e(t)}{BDP} + q_{ki}\int_0^t \frac{e(t)}{BDP}dt, \quad 0 \le p(t) \le 1 \tag{2}$$

Parameter selection: The parameters of RQ-AQM should be calculated according to the stability condition for given network conditions. Consider a network with C = 5625 packets/s (corresponding to 45 Mb sec^{-1} with packet size of 1000 bytes). According to stability condition (Sun and Zukerman, 2007), select r_{kp} = 0.0095, q_{kp} = 0.0077 and q_{ki} = 0.08. Then, from the simulation results, it can be seen that these parameters are able to stabilize the queue length around preset queue target.

According to the description of the RQ-AQM scheme, its parameters are adaptive to the bottleneck link capacity and the round-trip time. Former research results have shown that the stability of TCP/AQM System increases when the number of connections increases (Floyd and Jacobson, 1993). Therefore, when the parameters of RQ-AQM are selected for the unfavorable case where the number of connections is small, the superior stability of RQ-AQM will be fairly independent of the parameter setting and of traffic conditions.

LITERATURE REVIEW

AQM is aimed to reduce queue lengths and oscillations by maintaining high link utilization with fair resources allocation (Floyd *et al*., 2001). In other words, it is necessary to prevent buffer overflow or long queue length (Lu *et al*., 2005), otherwise the TCP flows would always stuck; meanwhile, buffer emptiness should be avoided, otherwise low link utilization would frequently occur. Correspondingly, AQM should be simple and scalable to be deployed in high-speed routers. Moreover, AQM should be stable and robust under dynamic environments or rather, the changing network parameters, such as the number of flows (N), Round-Trip Time (RTT), etc. (Lu *et al*., 2005). To achieve those goals, many AQM algorithms have been researched in the related literature. The principal differences among AQM proposals are the technique used to detect congestion and the control mechanism that stabilizes the queue length (Floyd *et al*., 2001). Here, only present several representatives in terms of their classifications in order for the comparative study through simulations.

AQM schemes based on queue length merit: The queue-based AQM algorithm uses the average (or instantaneous) queue length to compute the packet dropping/ marking probability. The well-known RED and its modifications (Hollot *et al*., 2002; Bonald *et al*., 2000; Sun *et al*., 2003a, b; Lin and Moms, 1997) belong to the category (Aweya *et al*., 2001).

Random Early Detection (RED) scheme: The RED algorithm was first proposed by Floyd and Jacobson (Yin and Low, 2001). It detects incipient congestion by using the average queue length as congestion indicator and comparing it against two thresholds min_{th} and max_{th}. Within this region, packets are randomly drops or marks based on using a linear probability function of the average queue size. Besides min_{th} and max_{th}, another two parameters are also required: a maximum drop probability P_{max} for early discard and an absorption factor used in the average queue size function ω_q. The detailed algorithms are demonstrated as follows.

First, low-pass filter is used to compute the average queue length (avg(t)).

$$avg(t) = (1-\omega_q)\times avg(t-1)+\omega_q \times q(t) \quad (3)$$

Where:

q(t) = The current queue length

ω_q = A weight parameter in the interval (0, 1)

Second, the packet marking/dropping probability p_a is computed:

$$p_a = \frac{p_b}{1-count\times p_b} \quad (4)$$

$$p_b = \max\left(0, p_{max}\times\frac{avg-min_{th}}{max_{th}-min_{th}}\right), \text{ if } avg < max_{th} \quad (5)$$

Where:

Count = The number of undropped packets since the last dropped packet

p_b = The temporary marking probability calculated based on heuristic observation

p_{max} = The maximum dropping/marking probability

As avg varies from min_{th} to max_{th} the packet-dropping/marking probability p_b varies linearly from 0 to p_{max}. If avg is less than the minimum threshold, no packets are dropped. If it exceeds the maximum threshold max_{th}, all incoming packets are dropped. If it is in between, p_b is increased linearly with the increasing avg.

Actually, RED keeps the average queue size low while allowing fluctuations in the actual queue size in order to accommodate bursty traffic and transient congestion. To avoid a bias against bursty traffic and the global synchronization that exists in drop-tail gateways, the RED gateway uses randomization to choose which arriving packets to drop. The probability of dropping a packet from a particular connection is roughly proportional to that connection's share of the bandwidth through the gateway. In other words, there are no precise rules for tuning those parameters (Wydrowski and Zukerman, 2002a, b) and most published results point at the difficulty of finding a robust RED configuration (for instance (Wydrowski and Zukerman, 2002a; Long *et al*., 2005). Kunniyur and Srikant (2001) observed that RED allows unfair bandwidth sharing when a mixture of different traffic shares a link. The unfairness comes from the fact that at any given time RED enforces the same drop rate upon all flows regardless of their bandwidth shares.

Adaptive random early detection scheme: So, to overcome RED's drawbacks, Hollot *et al*. (2002) proposed a self-configured RED algorithm. After that, Floyd (1994) proposed an Adaptive RED (ARED) algorithm based on the results of Feng obtained by Hollot *et al*. (2002). The basic idea is to modulate the aggressiveness of RED scheme by examining the variations in average queue length. If the avg is below min_{th}, the algorithm decreases if avg exceeds max_{th}, it increases p_{max}:

$$p_{max} \leftarrow P_{max}+\alpha, \text{ if avg>target and } p_{max} \le 0.5 \quad (6)$$

$$p_{max} \leftarrow P_{max}\times\beta, \text{ if avg<target and } p_{max} \ge 0.01 \quad (7)$$

It uses an additive-increase multiple-decrease policy to adjust p_{max} and p_{max} is adapted to keep the average queue size within a target range half way between min_{th} and max_{th}. The interval of target queue length target is defined in equation:

$$\begin{aligned} Target = [&min_{th}+0.4\times(max_{th}-min_{th}, \\ &min_{th}+0.6\times(max_{th}-min_{th}))] \end{aligned} \quad (8)$$

Through such mechanism, the adaptive RED algorithm dynamically adjusts the maximum dropping probability according to the network load and the average queue length within the bound of target queue length. And focusing on resolving the parameter sensitivity problem, ARED is supposed to improve the robustness of RED configuration.

Proportional-Integral controller scheme: Another approach to modify the packet drop probability function to improve the performances of RED and its robustness was proposed by Chang *et al.* (2004), named Proportional-Integral controller or PI controller. It responses quickly to the change of TCP/RED dynamics and some researchers also study new adaptive schemes to improve the performance of PI controller in recent years (Park *et al.*, 2004; Kunniyur and Srikant, 2000a, b).

The PI controller was developed based on the Linearized Fluid-based TCP Model (Bonald *et al.*, 2000). It marks each packet with a probability p(t) which is updated periodically using:

$$p(t) = p(t-1)+a(q(t)-q_{ref})-b(q(t-1)-q_{ref}) \quad (9)$$

where, a and b are constants. While this computation involves keeping two additional state variables, the computation requirement is not more than that of RED, since it gets the loss probability directly without need to obtain it via the loss profile using the average queue length. Using the linearized TCP/AQM dynamics, the PI controller not only improves responsiveness of TCP/AQM dynamics but also stabilizes the router queue length around the reference input q_{ref}, compared with original RED. However, the results also showed that it cannot achieve both satisfactory transient response and small deviation from steady-state behavior over a wide range of network dynamics.

This gave rise to new research on how to improve the performance of adaptive AQMs based on PI controller. Xu *et al.* (2005) proposed a novel nonlinear PI AQM algorithm which uses nonlinear functions of queue length error to adjust the control coefficients. Hong *et al.* (2004) and Athuraliya *et al.* (2001) proposed a self-tuning PI controller that adjusts the controller parameters when gain or phase margins falls outside a specified interval. Zhu *et al.* (2003) proposed an adaptive PI that increases the queue length error when it is larger than a certain threshold. Kunniyur and Srikant (2001) proposed to use average queue length and loss rate to adjust the drop probability thus increases the speed of the response of the PI controller. Though there are many adaptive PI proposed, their performance is still unclear (Park *et al.*, 2004).

AQM schemes based on load merit: Rate-based AQMs determine congestion and take actions based on the packet arrival rate. The goals of rate-based AQMs are to alleviate rate mismatch between enqueue and dequeue and to achieve low loss, low delay and high link utilization.

AVQ is one novel rate-based algorithm. It maintains a virtual queue whose capacity $\tilde{C}$ (called virtual capacity) is less than the actual link capacity C, i.e., $\tilde{C}=\beta C$ $(0<\beta<1)$. According to Hollot *et al.* (2001), when a packet arrives in the real queue, the virtual queue also updates to reflect the new arrival. And packets in the real queue are marked/dropped when the virtual buffer overflows. The virtual capacity at each link is then modified such that total flow entering each link achieves a desired utilization of the link. The virtual queue capacity is updated according to the following differential equation:

$$\dot{\tilde{C}}=\alpha(\gamma C - r(t)) \quad (10)$$

where, α and γ are control parameters which determine the convergence speed and the desired level of utilization, respectively; r (t) is the current input rate at the link. The implication behind is that marking has to be more aggressive when the link utilization exceeds the desired utilization and should be less aggressive when the link utilization is below the desired utilization.

A feature of the AVQ scheme that is appealing is in the absence of feedback delays, it is shown by Hollot *et al.* (2001) that the system is fair in the sense that it maximizes the sum of utilities of all the users in the network. It was shown by Hollot *et al.* (2001) that a fluid-model representation of the above scheme along with the congestion controllers at the end-hosts was semi-globally asymptotically stable when the update at the links was done sufficiently slowly.

AQM schemes based on concurrent queue and load merits: Different from other traditional AQM schemes, REM first introduced the concept of price as the measurement of congestion. The key idea is to decouple congestion measure (price) from performance measure (loss and queue) so that while congestion measure must vary with the number of sources, performance measure can be stabilized around its target independently.

The price functionis $\mu(t)$ defined based on the queuing information q(t), the queue length and the rate information r (t), the aggregate arrival rate at the link:

$$\mu(t) = \mu(t-1)+\gamma(\alpha(q(t-1)q_{ref}))+r(t-1)-C \quad (11)$$

where, γ and α are constants and q_{ref} is the target queue length. And at each link, it continuously updates the value of the price (the explanation of updating scheme seen by Wang *et al.* (2005) and marks packets with exponential probability. Then, an exponential marking probability function p(t) is formulated which gives the probability of packets being marked at the link at time t:

$$p(t) = 1-\phi^{-\mu(t)} \quad (12)$$

where, ϕ is an arbitrary constant greater than one. Like other AQM algorithms, REM also needs to address the problems of high link utilization, stable queue length and low packet loss. Floyd and Jacobson (1993) studied the local stable condition under the discrete time model but without considering feedback delay and stated that both high utilization and negligible loss and delay seems achievable without sacrificing the simplicity and scalability of the original RED. Global stability was proved for zero feedback delay in continuous and discrete time models, respectively (Jacobson, 1988; Ott *et al*., 1999). Yin and Low (2002) used the discrete time model and only presented the analytical result of one and twostep uniform feedback delay. Wang *et al.* (2005) by using a continuous time model in the multilink and multisource network, demonstrated that the local stability of REM depends on both the algorithm parameter settings and the network conditions.

SIMULATION AND COMPARISON

Stability test: The stability performance of RQ-AQM was first examined through simulation test. Whether the queue length can be stabilized at a target queue length is the key indicator to measure the stability performance. According to Li *et al.* (2006), if the queue length target can be closely achieved regardless of traffic conditions, the requirements of Quality of Service (QoS) for real time services could be satisfied.

The network topology used in the simulation is a single bottleneck network topology as in Fig. 1. The only bottleneck link is the common link between the two routers. The other links are assumed to have sufficient capacity to carry their traffic. The simulations were performed to validate the performance of RQ-AQM using ns2 (Reguera *et al*., 2008).

The parameters used are: the common link capacity is 45 Mb sec^{-1}, target queue length q_{ref} = 300, the packet size is 1000 bytes, the round-trip propagation delay is 100 msec, the buffer size is 1125 packets (twice the bandwidth-delay product of the network). Sample time interval: dt = 0.00625 sec, proportional gain of rate: r_{kp} = 0.0095, proportional gain of queue: q_{kp} = 0.0077 integral gain of queue: q_{ki} = 0.0005 (which corresponds to q_{ki} = 0.08 in continuous system when dt is 0.00625 sec). The primary performance metrics are the instantaneous queue length q(t).

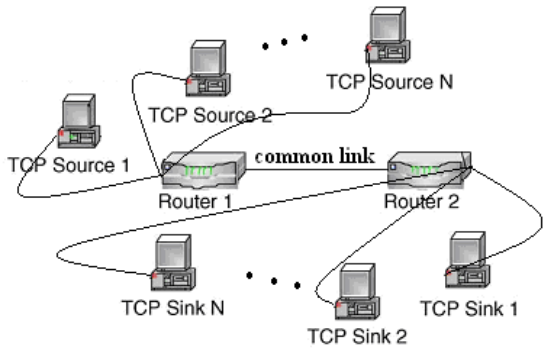

Fig. 1: Single bottleneck network topology for simulation

Case 1: Suppose there are various numbers of active TCP connections, i.e., 100, 500, 1000 and 1500 TCP connections. And all sources start data transmission at time 0. Figure 2 shows that the instantaneous queue

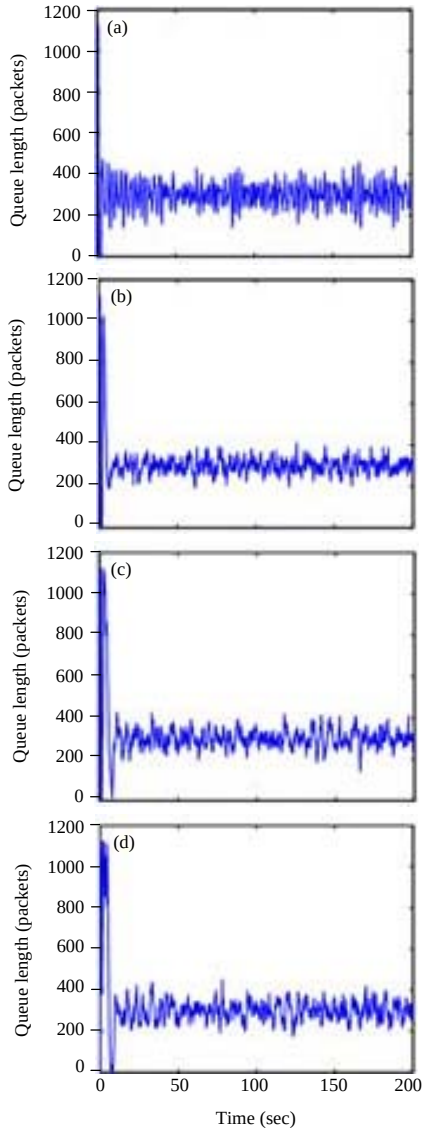

Fig. 2: Queue length variations for case 1: a) 100 TCP connections; b) 500 TCP connections; c) 1000 TCP connections; d) 1500 TCP connections

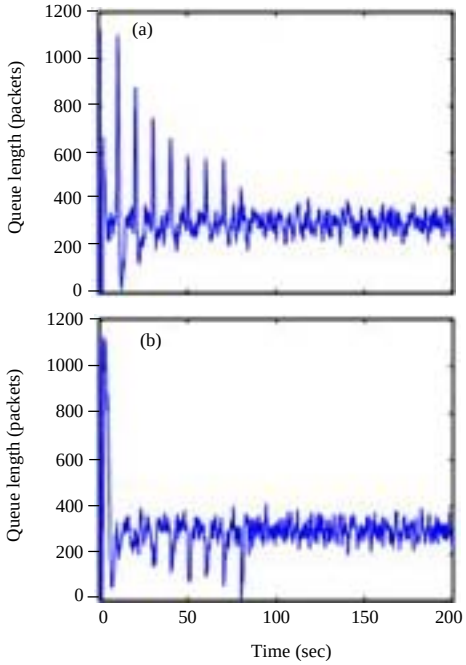

Fig. 3: The simulation results for case 2: a) number of TCP connections varies from 200-1000; b) number of TCP connections varies from 1000-200

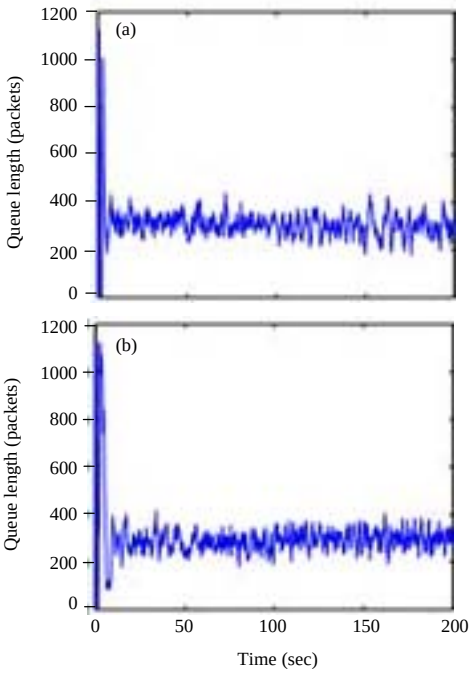

Fig. 4: Queue length variations for case 3: a) number of TCP connections varies from 200-1000; b) number of TCP connections varies from 1000-200

lengths q(t) of RQ-AQM are all kept around the target q_{ref} under 100, 500, 1000 and 1500 TCP connections.

Case 2: Suppose the number of TCP connections varies dynamically and uniformly start and stop. Here, ran two times simulation. First, the number of TCP connections varies from 200-1000. Second, the number of TCP connections varies from 1000-200. But in each of the runs, a group of 100 connections are started (or stopped) at the same time at each 10 sec interval. The results (Fig. 3) show that RQ-AQM is able to stabilize the queue length around the control target when the number of connections dynamically varies over time.

Case 3: Suppose the number of TCP connections varies dynamically but randomly start and stop. Two times simulation are also performed. First, the initial number of connections is set to 200 and the rest 800 connections have their start-time uniformly distributed over a period of 100 sec. Second, the initial number of connections is set to 1000, out of which 800 connections have their stop-time uniformly distributed over a period 100 sec. The results (Fig. 4) present that RQ-AQM is able to stabilize the queue length around the control target.

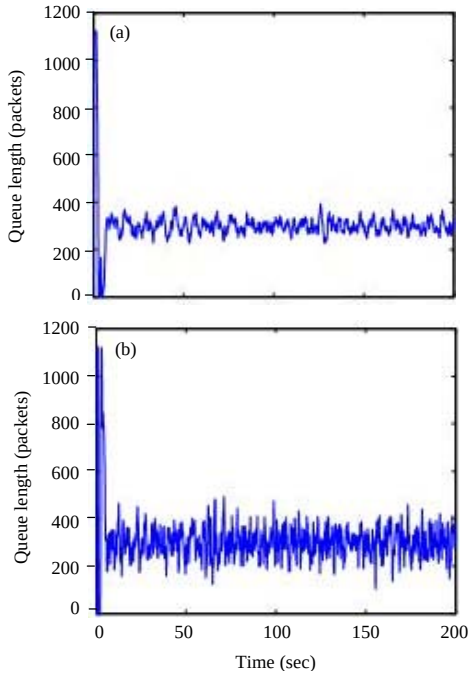

Fig. 5: Queue length variations for case 4: a) 45-15 Mb sec^{-1}; b) 45-115 Mb sec^{-1}

Case 4: Suppose under same TCP connections (i.e., 500), the link capacity varies from 45-15 and to 115 Mb sec^{-1}, respectively. The simulation results for 500 TCP connections in Fig. 5 also show that the system is stable.

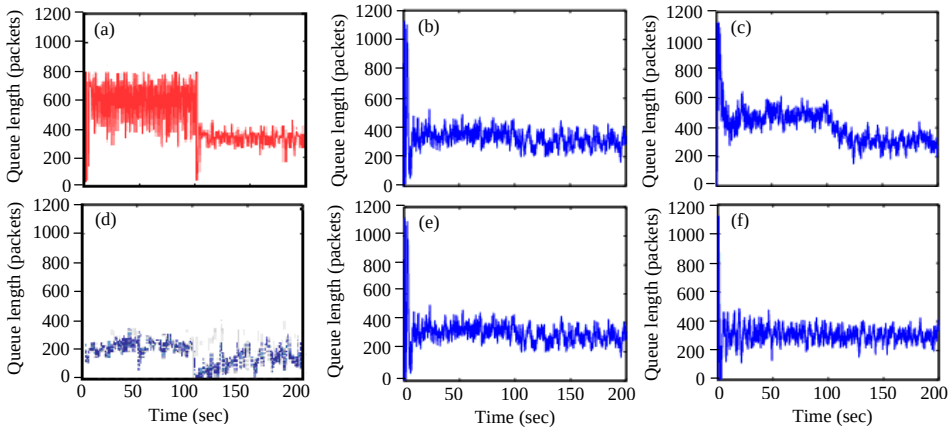

Fig. 6: Comparison of queue length variation for the six AQM schemes. a) RED, b) ARED, c) PI, d) AVQ, e) REM, f) RQ-AQM

Table 1: The parameters of RED, ARED, PI, AVQ, REM and RQ-AQM

RED	ARED	PI	AVQ	REM	RQ-AQM
$min_{th} = 0.25B$	$min_{th} = 15$	$a = 1.822\times10^{-5}$	$\alpha = 0.15$	$\alpha = 0.1$	$r_{kp} = 0.0095$
$max_{th} = 0.75B$	$max_{th} = 585$	$b = 1.816\times10^{-5}$	$\beta = 0.8$	$T_s = 0.01$	$q_{kp} = 0.0077$
$\omega_q = 0.02$	$\omega_q = 0.002$	$\omega = 160$	$\gamma = 1$	$\phi = 1.001$	$q_{ki} = 0.08$
$p_{max} = 0.1$	$\alpha = 0.01$	-	-	$\gamma = 0.001$	-
-	$\beta = 0.9$	-	-	-	-
-	Interval time = 0.5 sec	-	-	-	-

Table 2: Mean and standard deviation of the queue length and packet loss value for the six AQMs

AQMs	RED	ARED	PI	AVQ	REM	RQ-AQM
Mean	307.8	344.2	378.0	245.2	321.0	304.7
Standard deviation	73.9	55.7	124.0	66.2	55.8	43.7
Packet loss value	-	5856.0	1210.0	-	1857.0	831.0

Comparison with other AQM algorithms: This study performs simulations to compare the performance of RQ with RED, ARED, PI controller, AVQ and REM. The network topology used in the simulation is the same as Fig. 1. The related parameters are set as same as in stability test. Additionally, assume: the round-trip propagation delay of the TCP connections is uniformly distributed between 50 and 500 msec and marking the ECN bit is used for all the six AQM schemes.

For RED, set the parameters: $min_{th} = 0.25B$, $max_{th} = 0.75B$, $\omega_q = 0.02$ and $p_{max} = 0.1$. For ARED, set the parameters: $min_{th} = 15$, $max_{th} = 585$ and $\omega_q = 0.002$ and other parameters are set the same as (Wang *et al*., 2004): $\alpha = 0.01$, $\beta = 0.9$, interval time = 0.5 sec. For PI controller, use the default parameters in ns2: $a = 1.822\times10^{-5}$, $b = 1.816\times10^{-5}$ and the sampling frequency $\omega = 160$. For AVQ, set $\alpha = 0.15$, $\beta = 0.8$, $\gamma = 1$. For REM, the default parameters of Aweya *et al*. (2001) are used: $\alpha = 0.1$, $T_s = 0.01$, $\phi = 1.001$, $\gamma = 0.001$. For RQ-AQM. Table 1 summarizes all the parameters set for the six AQM schemes.

In the simulation, the initial number of connections is set to 200 and 800 additional connections have their start-time uniformly distributed over a period of 100 sec. Figure 6 shows the queue lengths for all six AQMs. It can be seen that RQ-AQM reacts and converges to the target queue length of 300 faster than all five others.

In order to evaluate the performance in steady-state, the average and the standard deviation of queue length at the last 150 sec are calculated. The results and packet loss value are shown in Table 2. It is observed that RQ-AQM queue length had the mean of 304.7 which is closest to the target of 300 and achieved the lowest standard deviation and lowest packets loss value compared with all other AQMs.

CONCLUSION

The primary goals of congestion control or AQM design (Paganini, 2001) are to predict incipient congestion timely and accurately with low queuing delays, high link utility, simplicity, stability and robustness to variation in traffic load. Recently, many researchers have demonstrated the inherent weakness of current rate-based AQM algorithms. Due to their complete separation from queue merit, subtly conservative and aggressive packet marking dynamic are normally caused. On the other hand, the queue-based AQM Method also can not survive from its the main drawback that is the control mechanism

inherently necessitated a backlog of packets which creates unnecessary large queuing delay and queuing delay jitter.

Therefore, this study has managed to develop a novel RQ-AQM scheme based on both input rate and queue length in order to achieve a better internet congestion control. Firstly, it designed the objective RQ-AQM algorithm and described its parameter selection. Then, other AQM algorithms were selectively introduced in three categories to form a theoretical foundation for the later on comparison. Thirdly, the stability of RQ-AQM was tested via simulation under several conditions. The results proved that RQ-AQM is able to maintain the queue length around the given target. Finally, the comparison with other AQM schemes demonstrated the superiority of RQ-AQM in low packet loss, faster convergence to target queue length and closest to the target queue length.

LIMITATIONS

The main limitation of this study is that all the simulations were merely under a single bottleneck network topology. Furthermore, the system stability was examined under just a few cases. Other cases like TCP connections mixed with HTTP connections or UDP flows, etc. can also be tested. And more important, the multiple bottleneck scenarios should be covered in the future work with more parameters set to enhance the proved results.

REFERENCES

Athuraliya, S.A., S.H. Low, V.H. Li and Q.H. Yin, 2001. REM: Active queue management. IEEE Network, 15: 48-53.

Aweya, J., M. Ouellette and D.Y. Montuno, 2001. A control theoretic approach to active queue management. Comput. Networks, 36: 203-235.

Bonald, T., M. May and J.C. Bolot, 2000. Analytic evaluation of RED performance. IEEE Ann. Joint Conf. Comput. Commun. Soc., 3: 1415-1424.

Braden, B., D. Clark, J. Crowcroft, B. Davie and S. Deering *et al*., 1998. RFC2309: Recommendations on queue management and congestion avoidance in the internet. http://opalsoft.net/qos/TCP-1020.htm.

Chang, X.L., J.K. Muppala and J.T. Yu, 2004. A robust nonlinear PI controller for improving AQM performance. Proceedings of the IEEE International Conference on Communications, Volume 4, June 20-24, 2004, Paris, France, pp: 2272-2276.

Feng, W., D. Kandlur, D. Saha and K. Shin, 1999. A self-configuring RED gateway. Proceedings of the IEEE 18th Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3, March 21-25, 1999, New York, USA., pp: 1320-1328.

Floyd, S. and V. Jacobson, 1993. Random early detection gateways for congestion avoidance. IEEE/ACM Transa. Networking, 1: 397-413.

Floyd, S., 1994. TCP and explicit congestion notification. ACM Comput. Commun. Rev., 24: 8-23.

Floyd, S., R. Gummadi and S. Shenker, 2001. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management. AT&T Center for Internet Research at ICSI. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.78.7450&rep=rep1&type=pdf.

Hollot, C.V., V. Misra, D. Towsley and W.B. Gong, 2001. On designing improved controllers for AQM routers supporting TCP flows. Proc. IEEE Commun. Comput. Soci., 3: 1726-1734.

Hollot, C.V., V. Misra, D. Towsley and W.B. Gong, 2002. Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Trans. Automat. Control, 47: 945-959.

Hong, Y., O.W.W. Yang and C. Huang, 2004. Self-tuning PI TCP flow controller for AQM routers with interval gain and phase margin assignment. Proceedings of the IEEE Global Telecommunications Conference, Volume 3, November 29-December 3, 2004, Dallas, TX., USA., pp: 1324-1328.

Jacobson, V., 1988. Congestion avoidance and control. ACM SIGCOMM Comput. Commun. Rev., 18: 314-329.

Kunniyur, S. and R. Srikant, 2000a. Analysis and design of an Adaptive Virtual Queue (AVQ) algorithm for active queue management. Assoc. Comput. Machinery Comput. Commun. Rev., 31: 123-134.

Kunniyur, S. and R. Srikant, 2000b. End-to-end congestion control: Utility functions random losses and ECN marks. Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, March 26-30, 2000, Tel Aviv, Israel, pp: 1323-1332.

Kunniyur, S. and R. Srikant, 2001. A time scale decomposition approach to adaptive ECN marking. Proceedings of the IEEE 20th Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3, April 22-26, 2001, Anchorage, AK., USA., pp: 1330-1339.

Li, Y., K.T. Ko and G.R. Chen, 2006. Stable parameters for PI-control AQM scheme. Electron. Lett., 42: 887-888.

Lin, D. and R. Moms, 1997. Dynamics of random early detection. Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, September 14-18, 1997, Cannes, France, pp: 127-137.

Long, C., B. Zhao, X. Guan and J. Yang, 2005. The Yellow active queue management algorithm. Comput. Networks, 47: 525-550.

Lu, X.C., M.J. Zhang and P.D. Zhu, 2005. An adaptive PI active queue management algorithm. J. Software, 16: 903-910.

Ott, T.J., T.V. Lakshman and L.H. Won, 1999. SRED: Stabilized RED. Proceedings of the IEEE Infocom, March, 1999, IEEE Computer Society, New York, USA., pp: 1346-1355.

Paganini, F., 2001. On the stability of optimization-based flow control. Proceedings of the American Control Conference, Volume 6, June 25-27, 2001, Arlington, VA., USA., pp: 4689-4694.

Park, E.C., H. Lim, K.J. Park and C.H. Choi, 2004. Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks. Comput. Networks, 44: 17-41.

Reguera, V.A., F.F.A. Paliza, W. Godoy Jr. and E.M.G. Fernandez, 2008. On the impact of active queue management on VoIP quality of service. Comput. Commun., 31: 73-87.

Sun, J. and M. Zukerman, 2007. RaQ: A robust active queue management scheme based on rate and queue length. Comput. Commun., 30: 1731-1741.

Sun, J., K.T. Ko, G. Chen, S. Chan and M. Zukerman, 2003a. PD-RED: To improve the performance of red. IEEE Commun. Lett., 7: 406-408.

Sun, J., G. Chen, K.T. Ko, S. Chan and M. Zukerman, 2003b. PD-controller: A new active queue management scheme. Proceedings of the IEEE Global Telecommunications Conference, Volume 6, December 1-5, 2003, San Francisco, CA., USA., pp: 3103-3107.

Sun, J., M. Zukerman and M. Palaniswami, 2007. A stable adaptive PI controller for AQM. Proceedings of the International Symposium on Communications and Information Technologies, October 17-19, 2007, Sydney, Australia, pp: 707-712.

Wang, C., B. Li and K. Sohraby, 2004. API: Adaptive proportional-integral algorithm for active queue management under dynamic environments. Proceedings of the Workshop on High Performance Switching and Routing, April 19-21, 2004, Phoenix, AZ., USA., pp: 51-55.

Wang, C., B. Li, K. Sohraby and Y. Peng, 2003. AFRED: An adaptive fuzzy-based control algorithm for active queue management. Proceedings of the 28th Annual IEEE Conference on Local Computer Networks, October 20-24, 2003, Bonn, Germany, pp: 12-20.

Wang, C., B. Li, Y.T. Hou, K. Sohraby and K. Long, 2005. A stable rate-based algorithm for active queue management. Comput. Commun., 28: 1731-1740.

Wydrowski, B. and M. Zukerman, 2002a. QoS in best-effort networks. IEEE Commun. Mag., 40: 44-49.

Wydrowski, B. and M. Zukerman, 2002b. GREEN: An active queue management algorithm for a self managed internet. Proceedings of the IEEE International Conference on Communications, Volume 4, April 28-May 2, 2002, New York, USA., pp: 2368-2372.

Xu, Y., J. Yang and Q. Du, 2005. Nonlinear PI active queue management based on hyperbolic secant functions. Proceedings of the 4th International Conference on Machine Learning and Cybernetics, Volume 2, August 18-21, 2005, Guangzhou, China, pp: 715-720.

Yin, Q. and S.H. Low, 2001. Convergence of REM flow control at a single link. IEEE Commun. Lett., 5: 119-121.

Yin, Q. and S.H. Low, 2002. On stability of REM algorithm with uniform delay. Proceedings of the IEEE Global Telecommunications Conference, Volume 3, November 17-21, 2002, Taipei, Taiwan, pp: 2649-2653.

Zhu, L., G. Cheng and N. Ansari, 2003. Local stable condition for random exponential marking. IEE Proc. Commun., 150: 367-370.