NEWQUE: A New Approach to Active Queue Management for TCP with ECN

V. Santhi† and A.M. Natarajan††,

†Lecturer, Department of CSE, PSG College of Technology, Coimbatore, Tamil Nadu, India
††Professor and CEO, Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India

Summary
Active Queue Management (AQM) can potentially reduce packet drop rate in the Internet. This is used by routers to control congestion, where packets are dropped before queue become full. In order to improve the performance of congestion routers, a new framework of AQM, namely NEWQUE active queue management algorithm supporting explicit congestion notification (ECN), is proposed. The objective of the new algorithm is to improve performance of congested routers by keeping low queuing delay, packet drop rate low, link utilization high, and link utilization stable. The NEWQUE AQM is implemented with help of ns2 simulator. The simulation shows that the proposed design outperforms the peer AQM schemes in terms of packet loss, link utilization and queuing delay.

Key words:
Active queue management, Congestion control, Explicit Congestion Notification (ECN), Queuing Delay, Packet Loss.

1. Introduction

Congestion in a network or internet creates observable problems for the end system: reduced availability, throughput and increases response times. When a packet is dropped before it reaches its destination, all of the resources it has consumed in transit are wasted. TCP congestion avoidance algorithms are used to prevent the congestion collapse of today's Internet. 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.

The Internet has mainly relied on the TCP congestion control in order to limit packet loss and fairly share network resources [2, 6]. However new applications are being deployed which do not use TCP congestion control and are not responsive to the congestion signals given by the network. Such applications are potentially dangerous because they drive up the packet loss rates in the network and can eventually cause congestion collapse. TCP congestion avoidance algorithms [8] alone are not suitable for controlling the congestion in the Internet. Some mechanisms are needed in the routers to give best performance to control congestion collapse. Improving the congestion control [8] and queue management algorithms in the Internet has been one of the most active areas of research.

There are two classes of mechanisms proposed by B. Braden et al. to congestion control at the router: "Queue management" and "Scheduling" algorithms [1]. In queue management algorithms manage the queue length by dropping packets when needed or appropriate, while scheduling algorithms determine which packet to send from the queue and mainly used for allocation of bandwidth among flows.

The traditional technique, "tail-drop", is used in the today Internet for managing router queue lengths is to set a maximum length (in terms of packets) for each queue, accept packets for the queue until the maximum length is reached, then drop subsequent incoming packets until the queue decreases because a packet from the queue has been transmitted. But it has two important drawbacks such as lock-out problems and always maintains the queue becomes full. To avoid this situation a new technique called Active Queue Management (AQM) implemented [1].

By dropping packets before buffers overflow, active queue management allows routers to control when and how many packets to drop. In summary, an active queue management mechanism can provide the advantages for responsive flows such as reduce number of packets dropped in routers, provide lower-delay interactive service and avoid lock-out behavior.

The aim of this paper to design a NEWQUE AQM based on total flow arrival rate, link capacity and link utilization. This algorithm is rate-based scheme to predict the congestion and take actions based on the packet arrival rate. It is evaluated using ns2 simulator. The simulation shows that the NEWQUE AQM outperforms other active queue management techniques like BLUE, PI, RED and REM in terms of percentage packet loss, low average queuing delay and better link utilization.

Manuscript received March 5, 2010
Manuscript revised March 20, 2010

The rest of the paper is organized as follows. Section 2 gives a description of some of the AQMs such as RED, REM, BLUE and PI. It shows how the related AQMs are managing congestion. Section 3 describes NEWQUE AQM and provides a detailed analysis. Section 4 describes evaluation of its performances based on simulations. Finally, Section 5 concludes with a discussion of future work.

2. Background

One of the biggest problems with TCP's congestion control algorithm over drop-tail queues is that sources reduce their transmission rates only after detecting packet loss due to queue overflow. Since a considerable amount of time may elapse between the packet drop at the router and its detection at the source, a large number of numbers of packets may be dropped as the senders continue transmission at a rate the network cannot support.

2.1 Random Early Detection (RED):

RED [7] starts to probabilistically drop packets long before the buffer is full, providing early congestion indication to flows which can then gracefully back off before the buffer overflows. RED maintains two buffer thresholds. When the weighted average queue size is smaller than the first threshold, no packet is dropped, and when the weighted average queue size is larger than the second threshold, all packets are dropped. When the weighted average queue size has between these two thresholds, the packets are dropped based on marking probability $maxp$. It is based on queue length as an estimator of congestion and also requires a wide range of RED parameters to operate correctly under different congestion scenarios. Unfortunately, when a large number of TCP sources are active, the aggregate traffic generated is extremely burst. Burst traffic often defeats the active queue management techniques used by RED since queue lengths grow and shrink rapidly. Since RED uses average queue length to determine the marking probability $maxp$, this implies that average queue length must steadily increase as number of sender increases. While ECN [4] is necessary for eliminating packet loss in the Internet, we show that RED, even though it is used with ECN, is ineffective in preventing packet loss.

2. 2 Random Exponential Marking (REM):

REM is a framework to converse congestion information from links to sources by exponential marking. It uses two key features match rate clear buffer and sum prices. Three different alternative pricing algorithms PC1-PC3 constitute REM. PC3 is evaluated here Equation (1), because of its superiority over PC1 and PC2 [9]:

$$\text{PC3}: p_l(t+1) = [p_l(t) + \gamma(\alpha_l b_l(t) + x_l(t) - c_l)]^+ \quad (1)$$

where x_l is the aggregate arrival rate, c_l link capacity, b_l backlog, and α_l and γ are pricing constants. An exponential function determines the marking probability from the link price Equation (2):

$$m = 1 - \varphi^{-p_l(t)} \quad (2)$$

where φ controls marking. Although REM works very well in a steady state situation, the behavior in transient conditions and with reasonably constrained buffer sizes is not necessarily optimal. The simulation results detail how in an environment with a wide variation in N and finite buffers the performance suffers in terms of percentage packet loss and average queuing delay.

2.3 BLUE:

Unlike RED, BLUE [15] uses packet loss and link utilization history to manage the congestion. The marking probability Pm is updated based on the configuration parameters $\partial 1$, $\partial 2$, $freeze\_time$. The simulation results detail that how it suffers performance with wide variation in N.

2.4 PI Controller:

PI Controller [14] is based on feedback control theory, whose marking probability is updated based on the queue length as Equation (3)

$$p(k+1) = p(k) + a(q(k+1) - q_{ref}) - b(q(k) - q_{ref}) \quad (3)$$

where a and b constants. It computes a new p every T seconds

It has been shown in [14] that the PI AQM scheme outperform RED in terms of system response and steady-state error. It has some limitations such as 1) the linearization introduces model error; 2) average queue length increases as number of senders increases; 3) it is mainly depends on q_{ref} .

3. NEWQUE AQM

In order to improve the performance of congested routers in terms of low queuing delay and low packet loss rate, a new queue management algorithm called NEWQUE AQM [11] implemented. It is rate-based AQM to take actions against congestion. It has been designed with the objective to 1) minimize packet loss rates, low queuing delay and improve link utilization; 2) avoid global synchronization of sources; 3) provide low average queue size to avoid lock-out behavior.

3.1 Proposed Mechanism:

NEWQUE uses the flow arrival rate, the link capacity and link utilization history to manage congestion rather than on the instantaneous or average queue lengths. Also, only a single marking probability is maintained, when the flow arrival rate is greater than or equal to the link capacity, this probability is incremented, and when the flow arrival rate is less than the link capacity, this probability is decremented and also when the link is idle, it is decremented. This effectively allows NEWQUE to learn the correct rate it needs to send back congestion notification. At the same time, the speed of updating of the marking probability depends on a parameter minTIVL. The following shows the NEWQUE algorithm. (Fig. 1).

First, we define the following parameters.

B	Buffer size
Q	Total queue length of active flows
C	Link Capacity
r_{new} (t)	Current Total Flow arrival Rate at router
r_{old} (t)	Previous Total Flow arrival Rate at router
P	Packet dropping or marking probability of flow
prevTime	Time when the previous update of P occurred
minTIVL	Minimum time interval between two successive updates of P
N	Total number of active flows
now	Current time

This NEWQUE AQM takes the following steps:

1) When a new packet of flow arrives at a router, the router calculates the value of EQ that represents Q plus the size of the arriving packet.
2) At particular time t, the router calculates the incoming total flow arrival rate using exponential averaging method. The following method shows the computation of arrival rate at the router. [5]
Computation of Total Flow arrival Rate: The rates r_{new} (t) are estimated at each router [5]. At each router, use exponential averaging with the parameter $e^{-T/K}$ to estimate the rate of flows. Eq (4) shows computation of total flow arrival rate.
Let t and l be the arrival time and length of the arriving packet, respectively. The estimated rate r_{new} (t), is updated every time a new packet is received.

$$r_{new\,(t)} = (1 - e^{-T/K})\, l / T + e^{-T/K}\, r_{old\,(t)} \qquad (4)$$

where,

T is the inter-arrival time between the current and the previous packet.
K is a Constant.
3) If r_{new} (t) < C then either enque or deque the packet and if r_{new} (t) >= C then either mark or drop the packet based on P.

Marking Probability P is updated as follows:
4) Upon flow arrival rate (r_{new} (t)) >= link capacity (C) or (EQ >= Buffer size (B)):
Increment the Marking Probability (P)
P is calculated as:
If ((now – prevTime) > minTIVL)
P = P + α;
prevTime = now;
5) Upon link idle event or flow arrival rate (r_{new}(t)) < link capcity (C):
Decrement the Marking Probability (P)
P is calculated as:
If ((now – prevTime) > minTIVL)
P = P - β
prevTime = now

Generally if (EQ > B), the router chooses drops the front packet from the buffer of the flow. If (EQ <= B), the router preferred the packet is ECN-capable, the router marks the first unmarked packet with probability P. If flow is non-ECN-capable, the router drops the front packet from the buffer with probability P.
We illustrate how our designed NEWQUE AQM can support ECN flows because ECN allows end-to-end notification of network congestion without dropping packets [3,12]. The ECN marking informs to senders to control sending rate when the buffer becomes full at the router. The advantages of using ECN are bandwidth up to bottleneck not wasted and no delay enforced by retransmission. The ECN is only useful when the buffer is not full. Otherwise the router has to drop the packet it wants to or not.
NEWQUE AQM can set a Congestion Experienced (CE) code point bit in the packet header instead of dropping the packet. The use of the CE code point with ECN allows the receiver to receive the packet, avoiding the potential for excessive delays due to retransmissions after packet losses. The TCP sender sets the ECN-capable transport (ECT) bit in the IP header of data packets to report to routers that it supports ECN. If the flow is ECN-capable (if ECT bit is set), the routers mark the first unmarked packet with probability P by setting the congestion experienced (CE) bit in the IP header. When the TCP receiver receives a packet with a CE bit set, it sets the ECN-Echo (ECE) flag

in the TCP header of the subsequent ACK packet. The receiver continues to set the ECE flag until it receives a data packet with the Congestion Window Reduced (CWR) flag set [12]. When the TCP sender receives an ACK packet with an ECE flag set, it reduces its congestion window size in half. In addition, the sender sets the CWR flag in the TCP header of the next data packet after the reduction of the congestion window size. The TCP end-systems react at most once per RTT to ACK packets whose ECE flag is set.

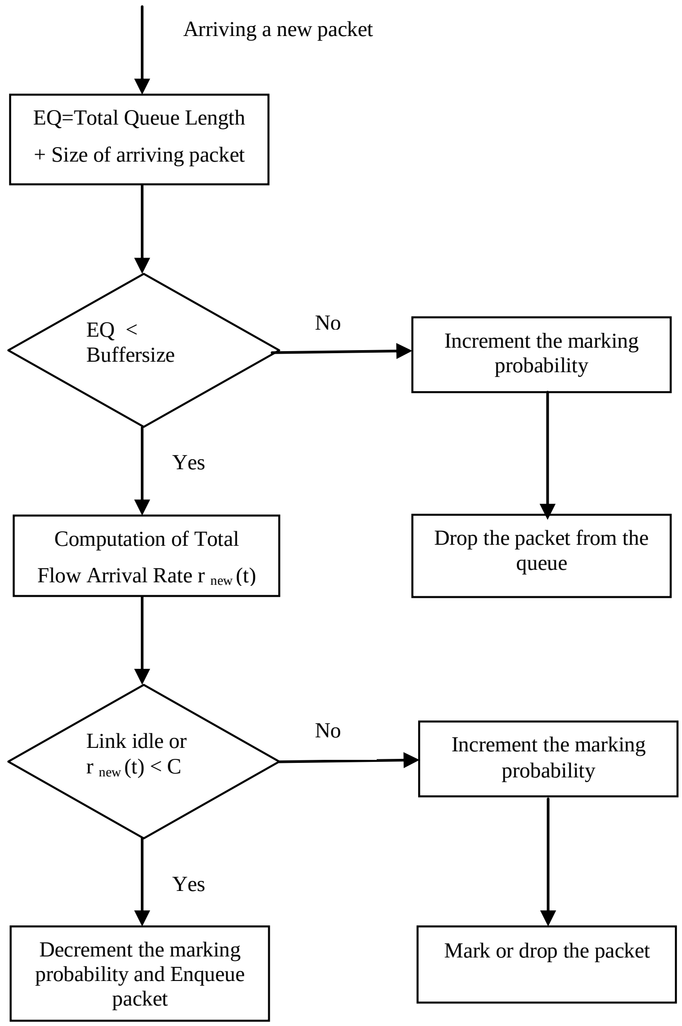

Fig. 1 A NEWQUE Algorithm

4. Simulations and Results

In order to evaluate the performance of NEWQUE, a number of simulation experiments are run using ns-2 [10] over a dumb-bell network topology shown in Fig. 2 and the multiple-bottleneck parking lot network topology depicted in Fig. 10 with varying number of input TCP connections.

To validate the performance of NEWQUE AQM [11] in terms of Link Utilization, Packet Loss Ratio, Average Queuing Delay, we conduct a simulation study in different scenarios. Some representative AQM schemes, namely, BLUE [15], RED [7], PI [14] and REM [9], are also simulated for the purpose of comparison.

4.1 Simulation Configuration

First, we consider the dumb-bell network topology depicted in Fig. 2, where number of TCP connections share a single bottleneck link. We assume that the TCP connection uses FTP sources, always have data to send. In addition, all FTP sources are enabled with ECN support. The links between the FTP sources and the router are 100 Mbps links with a 1 ms propagation delay, which are the same as those between the FTP sinks and the router. Router is connected to through a 10 Mbps 100 ms delay link. The maximum buffer size of each router is set to 300 packets. The packet size is 1040bytes.

The configuration parameters used in each of AQMs are: NEWQUE AQM: α=0.0001, β=0.02, minTIVL=100ms, K=0.1. In BLUE AQM: $\partial 1 = 0.025$, $\partial 2 = 0.0025$, freeze_time =100ms, which are recommended in [15]. In RED AQM: min_{th} = 20% of max buffer size (20% of 300 packets), max_{th} = 80% of max buffer size (80% of 300 packets). In PI AQM, we use the recommended values $a = 1.822 \times 10^{-5}$, b $= 1.816 \times 10^{-5}$ and $q_{ref} =$ 50% of max buffer size, given in [13]. In REM AQM: α=0.1, φ=1.001, and γ=0.001, which are recommended in [9]. The simulation time span is set to 100 seconds. Packet Loss statistics, Link Utilization and Average queuing delay are measured after 100 seconds.

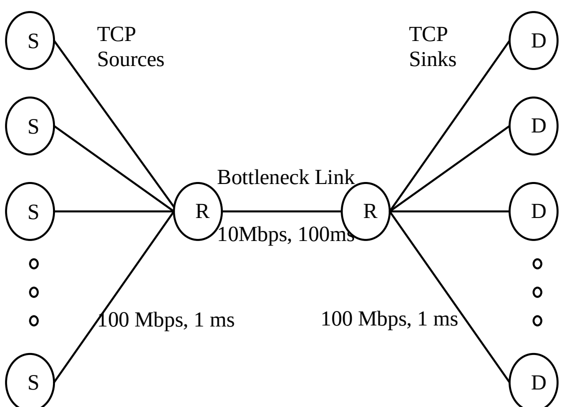

Fig. 2 Dumb-Bell Network topology

4.2 Scenario of Dumb-Bell Bottleneck Network Topology

1) Average Queue Size of Different AQM Schemes: In this simulation, the total numbers of TCP flows are varied from 50 to 300. The Buffer Size is fixed at 300 packets. Fig. 3 shows Average Queue Size (in packets) against number of TCP connections.

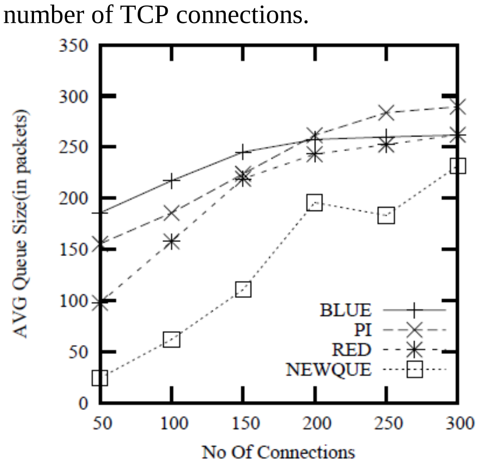

Fig. 3 Average Queue Size w.r.to Number of TCP flows

We can observe the NEWQUE AQM have smaller average queue size than other AQMs. By keeping the average queue size is small; it is challengeable for getting most of the incoming packets. It avoids Lock-out behavior. It reduces queuing delay also. The average queue lengths of REM and BLUE vary slightly with respect to the flow number, while the average queue of PI goes high when the flow number increases, which is due to the PI is highly dependent on the system parameters.

2) Average Queuing Delay of Different AQM Schemes: In this simulation, the total numbers of TCP flows are varied from 50 to 300. The buffer size is fixed at 300 packets. Fig. 4 shows that Average Queuing delay against the number of TCP connections. For each data packet, we measured the time from when it arrives at bottleneck router 1 until it has been transmitted from that router. We can see that NEWQUE AQM achieves lower average queuing delay than other AQMs like BLUE, PI and RED. Since NEWQUE AQM probabilistically drops packets before the buffer is full, the average queue length in NEWQUE AQM is lower than other. Fig. 5 shows Average Queuing Delay against the Buffer Size (in packets). The Buffer Size is varied from 150 packets to 400 packets. The total number of TCP connection is fixed at 200 sources.

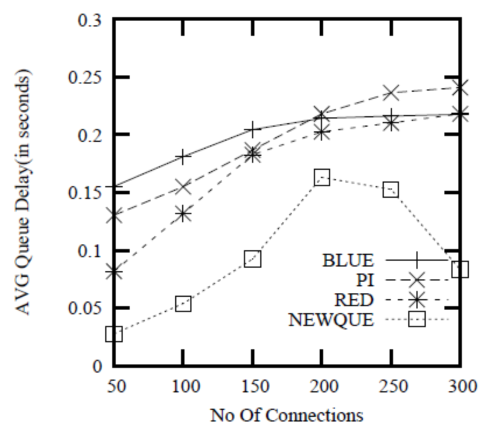

Fig. 4 Average Queuing Delay (in seconds) w.r.to Number of TCP flows

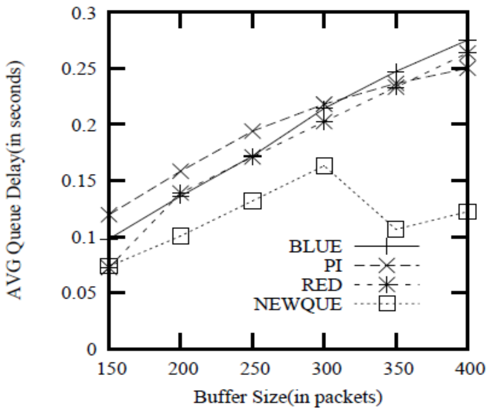

Fig. 5 Average Queuing Delay (in seconds) w.r.to Buffer Size (in packets)

Here we can observe that NEWQUE AQM has much the better performance in terms of average queuing delay regardless of buffer size. This is particularly important for web interactive applications whose performance is better when the end-to-end delay is low.

2) Percentage Packet Loss statistics of Different AQM Schemes: In this simulation, the total numbers of TCP flows are varied from 50 to 300. The buffer size is fixed at 300 packets. Fig. 6 shows that percentage packet loss against the number of TCP connections. We can observe that NEWQUE AQM has fewer drops rate (nearly less than 0.5%) than other AQMs like BLUE, PI, RED and REM. Fig. 7 shows Percentage Packet Loss against the Buffer Size (in packets). The Buffer Size is varied from 150 packets to 400 packets. The total number of TCP connection is fixed at 200 sources. Here we can observe that NEWQUE AQM has much the better performance in terms of percentage packet loss regardless of buffer size.

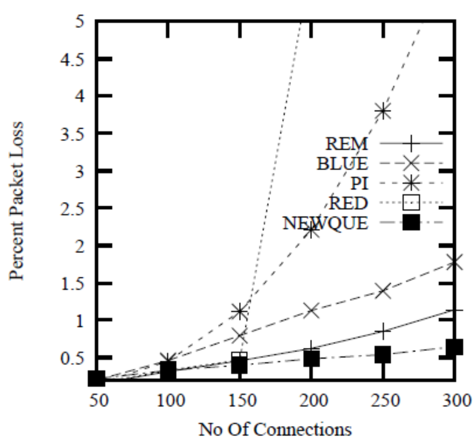

Fig. 6 Percentage Packet Loss w.r.to Number of TCP flows

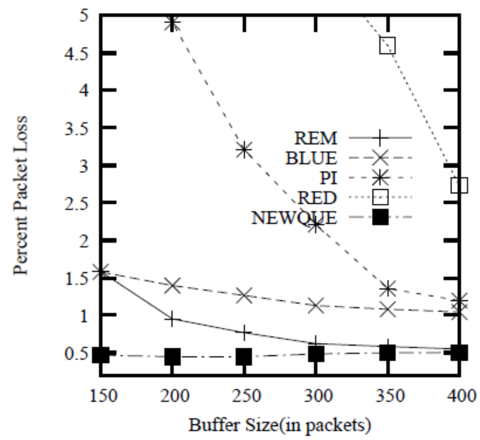

Fig. 7 Percentage Packet Loss w.r.to Buffer Size (in packets)

2) Percentage Link Utilization of Different AQM Schemes: In this simulation, the total numbers of TCP flows are varied from 50 to 300. The Buffer Size is fixed at 300 packets. Fig. 8 shows that Percentage Link Utilization against the number of TCP connections. Percentage Link Utilization is normalized by the bottleneck link capacity 10Mbps. We can observe that NEWQUE AQM have more than 99% of link utilization after the number of connection 150. Initially it gives lower percentage link utilization than other AQMs like BLUE, PI, RED but it reaches same level of link utilization after the connection 150. Fig. 9 shows Percentage Link Utilization against the Buffer Size (in packets). The Buffer Size is varied from 150 packets to 400 packets. The Number of TCP Connection is fixed at 200 Sources. Here we can observe that NEWQUE AQM have more than 99% Link Utilization regardless of the Buffer Size.

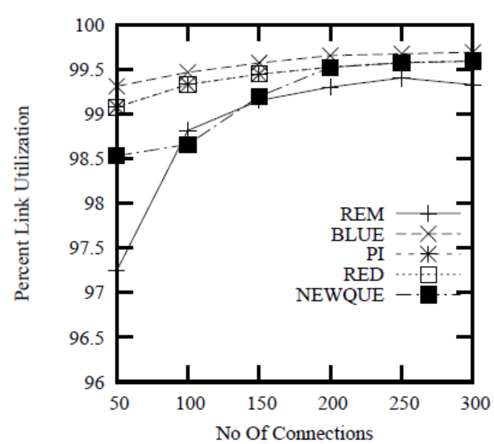

Fig 8 Percentage Link Utilization w.r.to Number of TCP flows

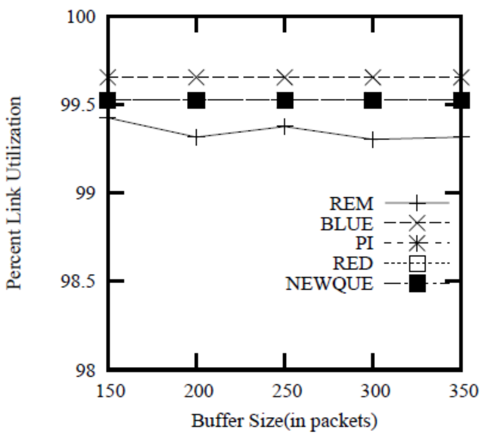

Fig 9 Percentage Link utilization w.r.to Buffer Size (in packets)

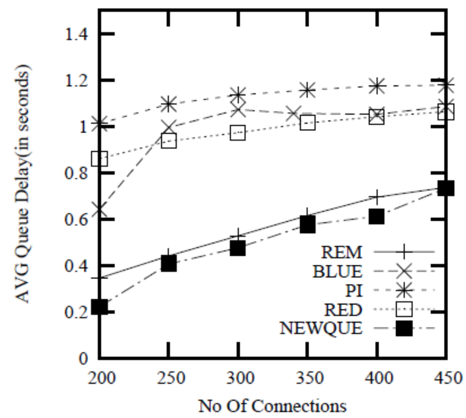

Fig. 11 Average Queuing Delay (in seconds) of Queue 3 w.r.to Number of TCP flows

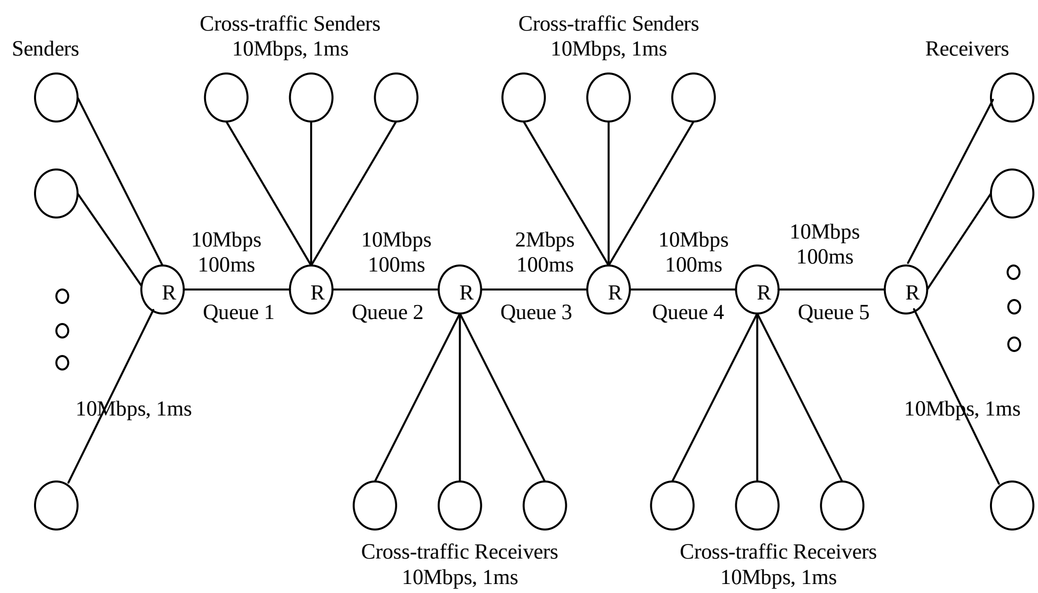

Fig. 10 Multiple-bottleneck parking lot network topology

4.3 Scenario of Multiple Bottleneck Topology

Using the multiple-bottleneck parking lot network topology depicted in Fig. 10, we study the performance of different AQM schemes in the presence of cross traffic. We set 300 TCP connections with sender at the left hand side and receivers at the right hand side, with 50 TCP flows for each cross traffic sender receiver pair and also the maximum buffer size of each router is 300 packets. The packet size is 1040 bytes.

The different performance metrics of Queue 3 are discussed. Fig. 11 and 12 shows the average queuing delay of Queue 3 against number of connections and against different buffer size (in packets). Fig. 13 and 14 shows the percentage packet loss of Queue 3 against number of connections and against buffer size (in packets). Fig. 15 and 16 shows the percentage link utilization of Queue 3 against number of connection and against different buffer size. Queue 2 and Queue 4 exhibit similar results of Queue 3 performances. Queue 1 and 5 are almost empty.

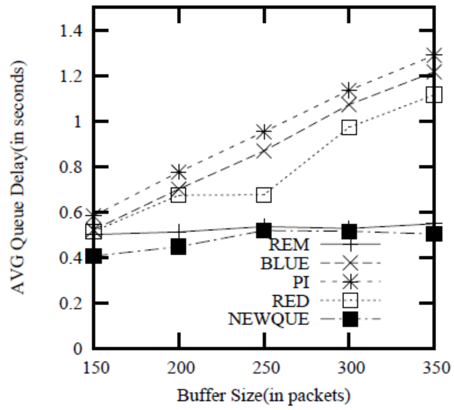

Fig. 12 Average Queuing Delay (in seconds) of Queue 3 w.r.to Buffer Size (in packets)

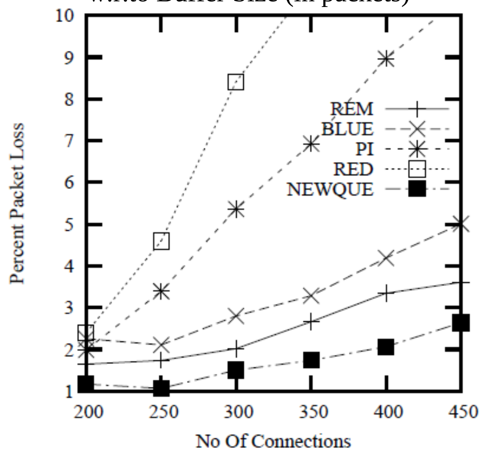

Fig. 13 Percentage Packet Loss of Queue 3 w.r.to Number of TCP flows

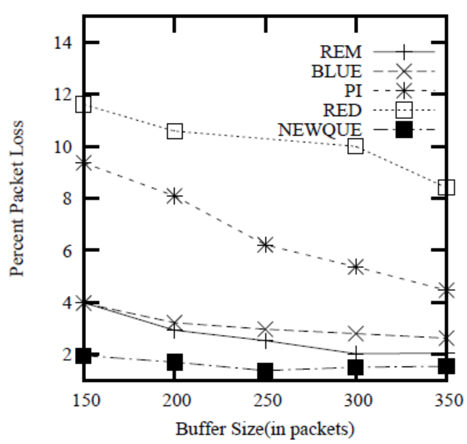

Fig. 14 Percentage Packet Loss of Queue 3 w.r.to Buffer Size (in packets)

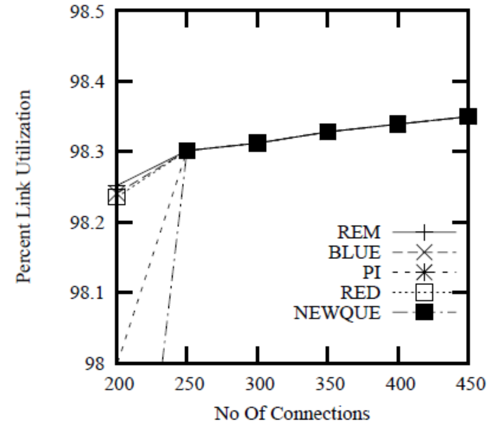

Fig. 15 Percentage Link Utilization of Queue 3 w.r.to Number of TCP flows

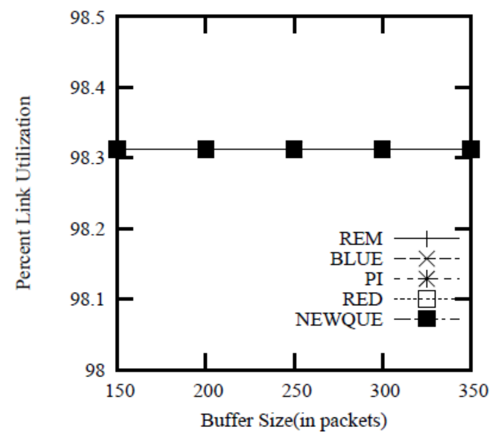

Fig. 16 Percentage Link Utilization of Queue 3 w.r.to Buffer Size (in packets)

From those figures, we can conclude that the NEWQUE AQM has better performance in terms of percentage packet loss, average queuing delay other than BLUE, RED, PI and REM. It has same percentage link utilization as such other AQMs. Similar results can be obtained under different TCP loads and different cross traffic loads.

5. Conclusion

In this paper, we developed a NEWQUE AQM scheme supporting ECN. It is simple, easily configured into a single router, making it easy to install. NEWQUE AQM shows that maintain low loss rates and low average queue size. The simulation experiments showed that the planned AQM scheme performs better than a number of AQM schemes in terms of percentage packet loss, percentage link utilization and average queuing delay. The proposed NEWQUE AQM is useful for web related applications. Finally, there are different areas in which such as heterogeneous round trip times, uncertain routing topologies and per-flow scheduling, the method presented here could be extended.

References

[1] R. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L.Peterson, K. Ramakrishnan, S.Shenker, J. Wroclawski, and L. Zhang, "Recommendations on queue management and congestion avoidance in the Internet", RFC 2309, Apr 1998.

[2] W. Feng, D. Kandlur, D. Saha, and K. G. Shin, "Techniques for eliminating packet loss in congested TCP/IP networks", Univ. Michigan, Ann Arbor, MI, Tech. Rep. UM CSE-TR-349-97, Oct. 1997.

[3] S. Floyd, "TCP and explicit congestion notification", Comput. Commun. Rev., vol. 24, no. 5, pp. 10-23, Oct. 1994.

[4] K. Ramakrishnan and S. Floyd, "A proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, Jan. 1999.

[5] Ion Stoica, Scott Shenkaer, and Fellow, "A Scalable Architecture to approximate fair bandwidth allocation in high speed networks", IEEE/ACM Transactions on networking, vol. 11, No. 1 February 2003.

[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] S.Floyd and V. Jacobson, "Random early detection for congestion avoidance", IEEE/ACM Transactions Networking, Vol. 1, pp 397-413, July 1993.

[8] V.Jacobson,"Congestion avoidance and control", in Proc. ACM SIGCOMM, Aug 1998, pp. 314-329.

[9] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, May-Jun. 2001.

[10] The network simulator ns-2 (2000). [Online]. Available: http://www.isi.edu/nsnam/ns/

[11] V. Santhi, Dr. A.M. Natarajan, "A New Approach to Active Queue Management for TCP with ECN", in Proc. International conference on Advanced Computing(ICAC 2009), Anna University Chennai, IEEE Madras Section, Chennai, Dec 2009, pp. 76-81.

[12] K. Ramakrishnan, S. Floyd and D. Black, "The addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, Sep. 2001.

[13] C. V. Hollot, Vishal Misra, Donald Towsley, and Weibo Gong "On designing improved controllers for AQM routers supporting TCP flows," in *Proc. IEEE INFOCOM*, Alaska, Apr. 2001, pp. 1726–1734.

[14] C. V. Hollot, Vishal Misra, Donald Towsley, and Weibo Gong "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. Autom. Control*, vol. 47, no. 6, pp. 945–959, Jun. 2002.

[15] W. Feng. D.Kadlur, D. Saha, and K. Shin, "The Blue active queue management", IEEE/ACM transactions on networking, vol 10, No.4, August 2002.

Dr.A.M.Natarajan is the CEO & Professor at Bannari Amman Institute of Technology, Sathyamangalam, Coimbatore. He received the B.E. degree in Electrical Engineering, and M.Sc.(Engg.) in Applied Electronics and Servo Mechanism from Madras University and Ph.D degree in System Engineering from Madras University. He has 40 years of teaching experience. He received the "Best Engineering college principal award in India for the Year 2000**"** from Indian Society for Technical Education, New Delhi. He has published more than 80 papers in International and National journals and conference proceedings. He has published 10 books. His areas of research include systems engineering and computer networks.

V. Santhi received the B.E. degree in Computer Science and Engineering from Madras University with distinction. She received M.E. degree in Computer Science and Engineering from Anna University Chennai with distinction. She has 7 years of teaching experience. She is currently pursuing doctoral research in computer networks management. At present she is working as Lecturer in Computer Science and Engineering Department at PSG College of Technology, Coimbatore.