

MNEWQUE: A New Approach to TCP/AQM with ECN

V. Santhi

Lecturer, Department of CSE
PSG College of Technology
Coimbatore, Tamil Nadu, India

A.M. Natarajan

Professor and CEO
Bannari Amman Institute of Technology
Sathyamangalam, Tamil Nadu, India

ABSTRACT

In order to improve the performance of congested routers, an Active Queue Management (AQM) is proposed. AQM can potentially reduce packet loss rate in the Internet. This is used to control congestion at the router, where packets are dropped before queue become full. A new framework of AQM, namely Modified NEWQUE (MNEWQUE) active queue management algorithm supporting explicit congestion notification (ECN), is developed by changing constant factor K in NEWQUE AQM. 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 MNEWQUE AQM is implemented with help of ns2 simulator. The simulation shows that the proposed design outperforms other AQM methods in terms of queuing delay, packet loss, and link utilization.

Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance Analysis and Design Aids. I.6 [Simulation and Modeling] – simulation output analysis.

General Terms

Algorithms, Performance evaluation.

Keywords

Active queue management, Congestion control, Explicit Congestion Notification (ECN), Queuing Delay, Link Utilization.

1. INTRODUCTION

The Internet Protocol Architecture is based on a connectionless end-to-end packet service using IP Protocol. The advantage of using this architecture provides flexibility and robustness. It needs careful design to provide good service under heavy load. In fact, lack of attention for forwarding packet at dynamic time could lead severe service degradation in the network. This is called as congestion collapse or internet meltdown [2]. Congestion in network generates severe 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. To overcome this problem, TCP congestion avoidance algorithms [5,7] recommended by Jacobson, are used to prevent the congestion collapse of today's Internet. These algorithms are operated in all hosts in the network to back off during congestion.

TCP congestion avoidance algorithms [5,7] 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.

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 dropping most recently added packet when the buffer is full. 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. The network delay is the combination of queuing delay and propagation delay. By reducing queuing delay could lead to reduces Round Trip Time (RTT). 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 Internet Engineering Task Force (IETF) introduces Explicit Congestion Notification (ECN) [3, 11] mechanism recommended by Ramakrishnan et al., in IP header to indicate the congestion from receiver to sender by packet marking instead of packet dropping. Any AQM router supports ECN reduces packet loss rate comparatively AQM router without ECN.

The aim of this paper to design a MNEWQUE AQM with ECN based on total flow arrival rate, and link capacity. 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 MNEWQUE AQM outperforms other active queue management techniques like BLUE, PI, RED and NEWQUE in terms of percentage packet loss, low average queuing delay and better link utilization.

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

2. OVERVIEW OF OTHER AQM METHODS

In this section provides brief introduction about different AQM methods with performance metrics, which are compared with MNEWQUE method.

2.1 Random Early Detection (RED)

RED [6] drops the packet from the buffer probabilistically before the buffer full. 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 max_p . 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. Since RED uses average queue length to determine the marking probability max_p , this implies that average queue length must steadily increase as number of sender increases. While ECN [3] 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 BLUE

Unlike RED, BLUE [14] uses packet loss and link utilization history to manage the congestion. The marking probability P_m 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.

2.3 PI

PI [13] is based on feedback control theory, whose marking probability is updated based on the queue length as Equation (1).

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

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

It has been shown in [13] 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} .

2.4 NEWQUE

NEWQUE AQM [9, 10] uses the flow arrival rate, the link capacity and link utilization history to manage congestion. Here the flow arrival rate is calculated using fixed weight exponential averaging method. The following Equation (2) is used to estimate the incoming arrival rate. Let t and l be the arrival time and length of the packet of flow.

$$r_{new}(t) = (1 - e^{-T/K})l/T + e^{-T/K}r_{old}(t)$$
 (2)

where T is the inter-arrival time between the current and the previous packet, $K=0.1$. It maintains single marking probability p, 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. Equation (2) shows computation of flow arrival rate. Here the different constant K value gives different flow arrival rate. To overcome this problem, MNEWQUE uses weight adaptive exponential averaging method.

3. Modified NEWQUE (MNEWQUE AQM)

The motivation behind Modified NEWQUE AQM (MNEWQUE) is to modify NEWQUE [9, 10] by changing constant factor K. The performance of MNEWQUE is improved in the network parameters like average queuing delay and low packet loss rate. It is rate-based AQM to take actions against congestion. It maintains two buffer threshold values such as max_{th} and min_{th} . The incoming rate of bottleneck router is estimated with help of these threshold values.

3.1 Proposed Algorithm

MNEWQUE uses the flow arrival rate, the link capacity to manage congestion. 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 also current queue length is greater than or equal to the buffer size, the probability is incremented. When the flow arrival rate is less than the link capacity, this probability is decremented. This effectively allows MNEWQUE 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 MNEWQUE algorithm.

First, we define the following parameters.

B	Buffer size
Q	Total queue length of active flows
C	Link Capacity
min_{th}	minimum buffer threshold
max_{th}	maximum buffer threshold
$r_{new}(t)$	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
now	Current time

The 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 weight adaptive exponential averaging method. The following method shows the computation of flow arrival rate. [4]

NEWQUE [9, 10] uses fixed weight adaptive exponential averaging method to estimate the incoming flows. From this we found that different constant values of K could lead to different

estimation values. In order to solve this problem, use weight adaptive exponential averaging method.

Computation of Total Flow arrival Rate: The rates $r_{new}(t)$ are estimated at each router [4]. At each router, use exponential averaging with the parameter $e^{-T/K}$ to estimate the rate of flows. Here we give the weight age to K. Equation (3) shows computation of total flow arrival rate.

Let t and l be the arrival time and length of the packet of flow. The estimated rate $r_{new}(t)$, is updated every time a new packet is received.

$$r_{new}(t) = (1 - e^{-T/K})l/T + e^{-T/K}r_{old}(t) \quad (3)$$

where T is the inter-arrival time between the current and the previous packet. K is calculated based on the following steps:

- i) If current queue length (Q) is less than or equal to 98% of maximum buffer threshold (max_{th})

$$K = k1 * \frac{max_{th} - Q}{max_{th} - min_{th}}$$

where $k1 = 0.1 * \frac{B}{C}$

- ii) If current queue length (Q) is greater than 98% of maximum buffer threshold (max_{th})

$$K = 0.001$$

- 3) If $r_{new}(t) \geq 95\%$ of C then either mark or drop the packet based on P otherwise enqueue the packet at the end of the queue.

Marking Probability P is updated as follows:

- 4) Upon flow arrival rate ($r_{new}(t)$) $\geq 95\%$ of link capacity (C) or ($EQ \geq$ Buffer size):

Increment the Marking Probability (P)

P is calculated as:

If ($(now - prevTime) > minTIVL$)

$$P = P + \alpha;$$

$$prevTime = now;$$

- 5) Upon flow arrival rate ($r_{new}(t)$) $<$ link capacity (C):

Decrement the Marking Probability (P)

P is calculated as:

If ($(now - prevTime) > minTIVL$)

$$P = P - \beta$$

$$prevTime = now$$

Generally if ($EQ \geq B$), 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 MNEWQUE AQM can support Explicit Congestion Notification (ECN) [3, 11] flows because ECN allows end-to-end notification of network congestion

instead of dropping packets. If the packet is ECN capable then it reduces packet loss rate in the network.

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. If the flow is set with ECN-capable transport (ECT) bit set, the MNEWQUE AQM mark the first unmarked packet with probability P and also set Congestion Experience (CE) bit in the IP header. If the CE bit is set by the router AQM then the TCP sinks react with setting of ECN-Echo (ECE) flag in the TCP header [11]. This information is passed to TCP senders through acknowledge packet given by the TCP sinks. Immediately the TCP senders react with reducing its incoming rate by setting the Congestion window Reduced (CWR) flag in the TCP header.

4. SIMULATIONS AND RESULTS

We simulate MNEWQUE AQM with ns-2 [8] under different network parameters over a dumb-bell network topology shown in Figure 1 and the multiple-bottleneck parking lot network topology depicted in Figure 10 with varying number of input active TCP connections. The performance of MNEWQUE AQM is validated in terms of percentage link utilization, percentage packet loss, and average queuing delay in different scenarios. Some representative AQM schemes, namely, BLUE [14], RED [6], PI [13] and NEWQUE [9, 10], are also simulated for the purpose of comparison.

4.1 Simulation Configuration

First, we use the dumb-bell network topologic structure depicted in Figure 1. S 's are number of active TCP sources. D 's are number of active TCP sinks. R 's are bottleneck routers. We assume that the TCP connection uses FTP application, always have data to send. In addition, all FTP packets are enabled with ECN support. The links between the TCP sources (S 's) and the router (R) are 100 Mbps links with a 1 ms propagation delay, which are the same as those between the TCP sinks (D 's) and the router (R). Router is connected to through a 10 Mbps 100 ms delay link, which is the bottleneck 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: MNEWQUE: $min_{th} = 20\%$ of max buffer size (300 packets), $max_{th} = 80\%$ of max buffer size (300 packets), $\alpha = 0.0001$, $\beta = 0.01$, $minTIVL=100ms$. In BLUE AQM: $\delta1 = 0.025$, $\delta2 = 0.0025$, $freeze_time = 100ms$, which are recommended in [14]. In RED AQM: $min_{th} = 20\%$ of max buffer size (300 packets), $max_{th} = 80\%$ of max buffer size (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 [12]. In NEWQUE AQM, we use: $\alpha = 0.0001$, $\beta = 0.01$, $minTIVL=100ms$, $K=0.1$ [9, 10]. The simulation time is set to 100 seconds. Packet Loss statistics, Link Utilization and Average queuing delay are measured after 100 seconds.

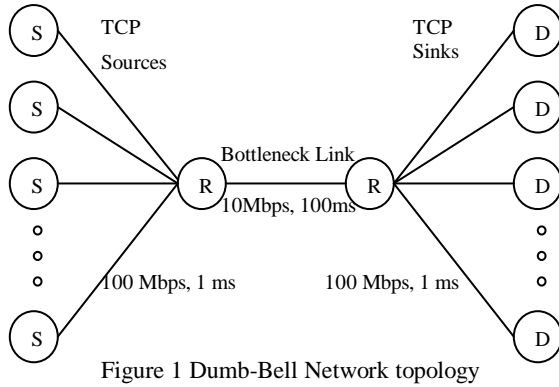


Figure 1 Dumb-Bell Network topology

4.2 Scenario of Dumb-Bell Bottleneck Network Topology

4.2.1 Average Queue Size of Different AQM Schemes:

In this simulation, the total numbers of TCP flows are varied from 100 to 300. The Buffer Size is fixed at 300 packets. Figure 2 shows Average Queue Size (in packets) against number of TCP connections. Here, we can see that the MNEWQUE 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 RED 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. But the NEWQUE is better than RED, BLUE and PI unlike MNEWQUE AQM. Figure 3 shows Average Queue Size (in packets) over different Buffer Size. The total number of TCP connection is fixed at 200 sources. We can also observe that MNEWQUE AQM have smaller average queue size than other AQMs.

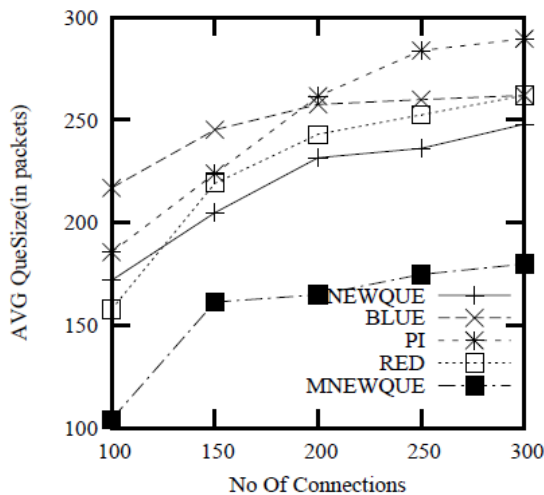


Figure 2 Average Queue Size w.r.to Number of TCP flows

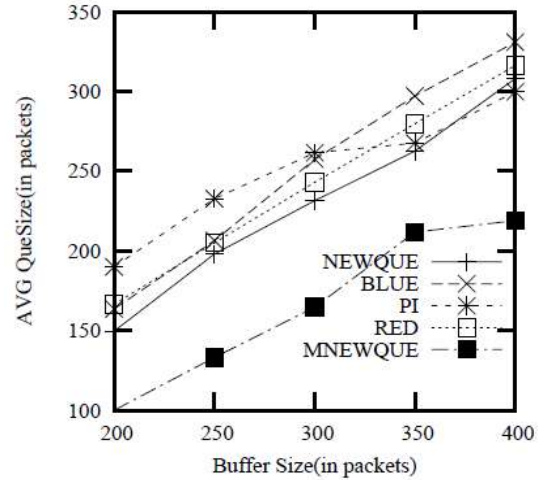


Figure 3 Average Queue Size w.r.to Buffer Size (in packets)

4.2.2 Average Queuing Delay of Different AQM Schemes:

In this simulation, the total numbers of TCP flows are varied from 100 to 300. The buffer size is fixed at 300 packets. Figure 4 shows that Average Queuing delay over 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 MNEWQUE AQM achieves lower average queuing delay than other AQMs like BLUE, PI, RED and NEWQUE. Since MNEWQUE AQM probabilistically drops packets before the buffer is full, the average queue length in MNEWQUE AQM is lower than other. Figure 5 shows Average Queuing Delay against the Buffer Size (in packets). The Buffer Size is varied from 200 packets to 400 packets. The total number of TCP connection is fixed at 200 sources. Here we can see that MNEWQUE AQM has much the better performance in terms of average queuing delay regardless of buffer size. This is particularly important for web interactive applications like short web transfers, telnet traffic whose performance is better when the end-to-end delay is low.

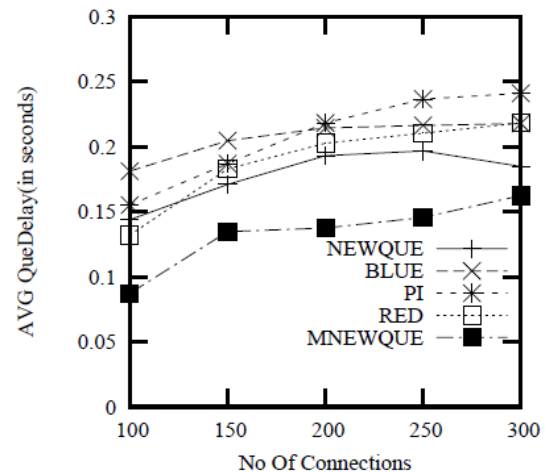


Figure 4 Average Queuing Delay (in seconds) w.r.to Number of TCP flows

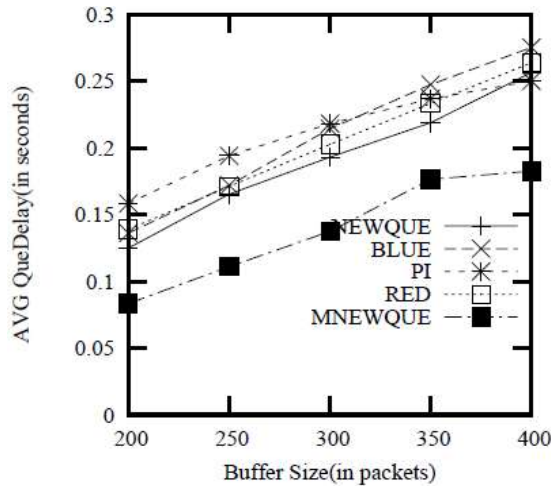


Figure 5 Average Queuing Delay (in seconds) w.r.to Buffer Size (in packets)

4.2.3 Percentage Packet Loss statistics of Different AQM Schemes:

In this simulation, the total numbers of TCP flows are varied from 100 to 300. The buffer size is fixed at 300 packets. Figure 6 shows that percentage packet loss against the number of TCP connections. We can observe that MNEWQUE AQM has fewer drops rate (nearly less than 0.3%) than other AQMs like BLUE, PI and RED. The percentage packet loss statistics of PI and RED are increases with increasing number of input TCP connections. Figure 7 shows Percentage Packet Loss against the Buffer Size (in packets). The Buffer Size is varied from 200 packets to 400 packets. The total number of TCP connection is fixed at 200 sources. Here we can observe that MNEWQUE AQM has much the better performance in terms of percentage packet loss regardless of buffer size.

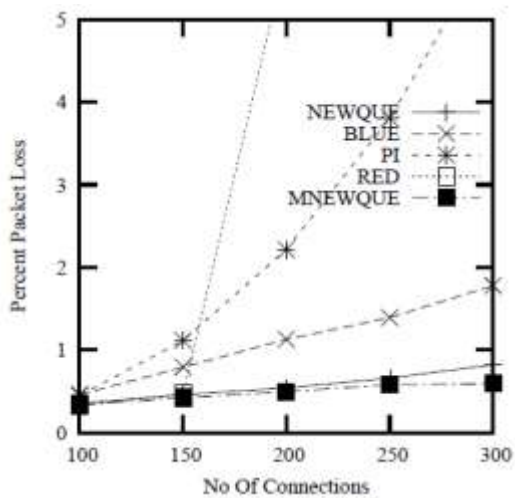


Figure 6 Percentage Packet Loss w.r.to Number of TCP flows

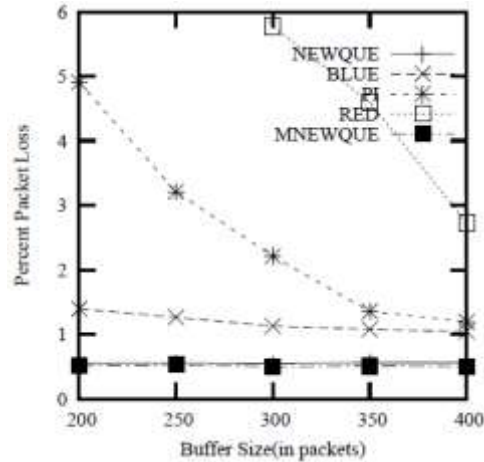


Figure 7 Percentage Packet Loss w.r.to Buffer Size (in packets)

4.2.4 Percentage Link Utilization of Different AQM Schemes:

In this simulation, the total numbers of TCP flows are varied from 100 to 300. The Buffer Size is fixed at 300 packets. Figure 8 shows that Percentage Link Utilization over the number of TCP connections. Percentage Link Utilization is normalized by the bottleneck link capacity 10Mbps. We can observe that MNEWQUE AQM have more than 99% of link utilization. Figure 9 shows Percentage Link Utilization against the Buffer Size (in packets). The Buffer Size is varied from 200 packets to 400 packets. The Number of TCP Connection is fixed at 200 sources. Here we can observe that MNEWQUE AQM have more than 99% Link Utilization regardless of the Buffer Size.

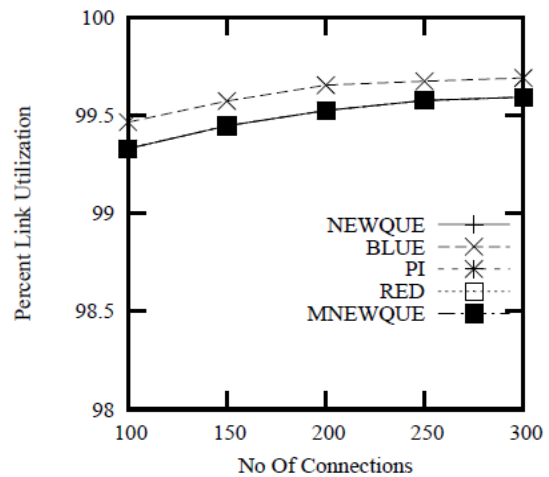


Figure 8 Percentage Link Utilization w.r.to Number of TCP flows

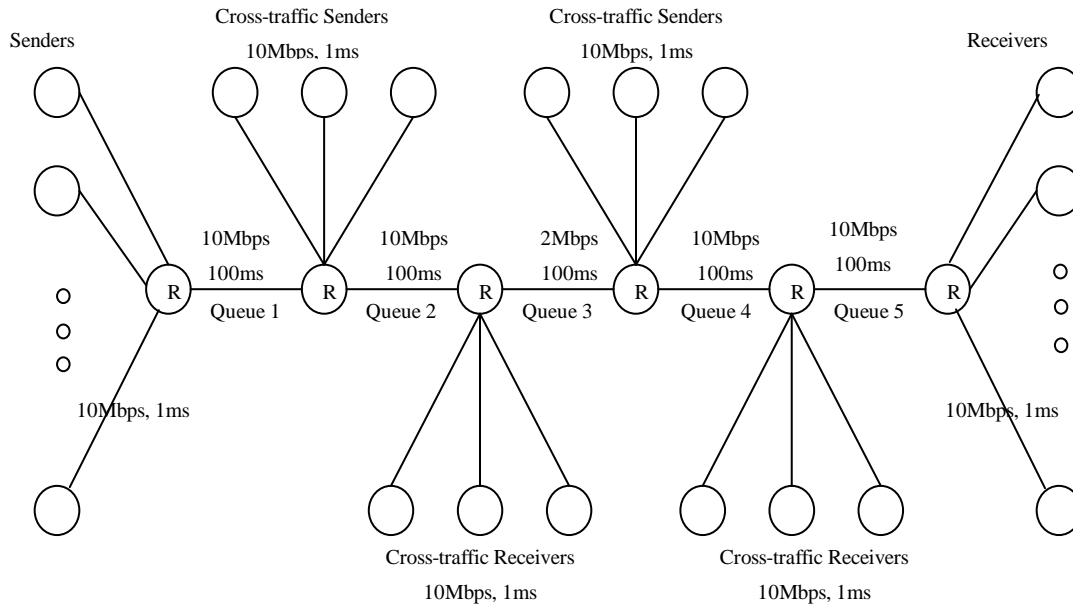


Figure 10 Multiple-bottleneck parking lot network topology

4.3 Scenario of Multiple Bottleneck parking lot Network Topology

Using the multiple-bottleneck parking lot network topologic structure depicted in Figure 10, we also conduct the simulations based on this network with two bottleneck links. Here we study the performance of different AQM schemes in the presence of cross traffic. We set varying number of TCP connections with sender at the left hand side and receivers at the right hand side, with fixed 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 simulation time span is 0-75 seconds. The results are taken after 75 seconds.

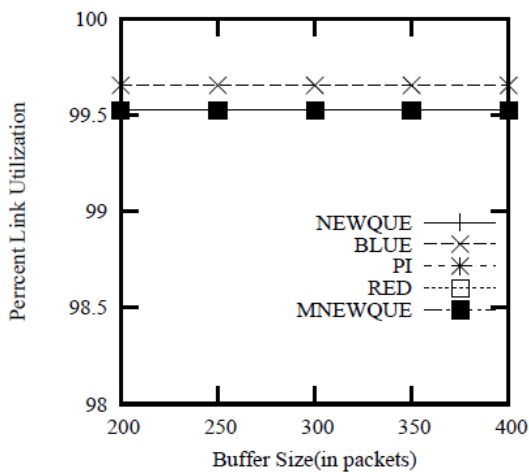


Figure 9 Percentage Link utilization w.r.to Buffer Size (in packets)

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

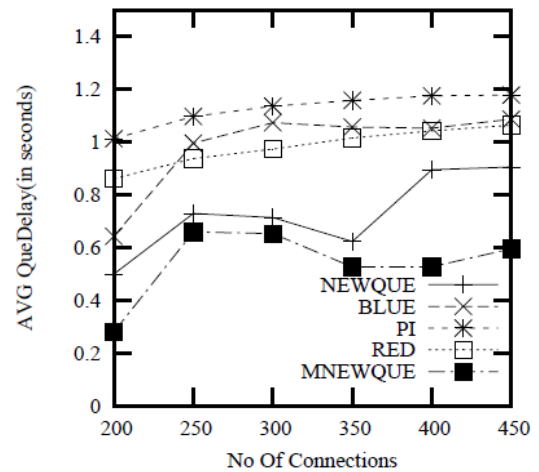


Figure 11 Average Queuing Delay (in seconds) of Queue 3 w.r.to Number of TCP flows

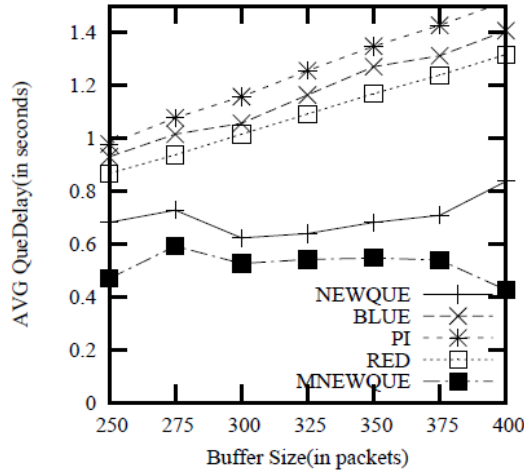


Figure 12 Average Queuing Delay (in seconds) of Queue 3 w.r.to Buffer Size (in packets)

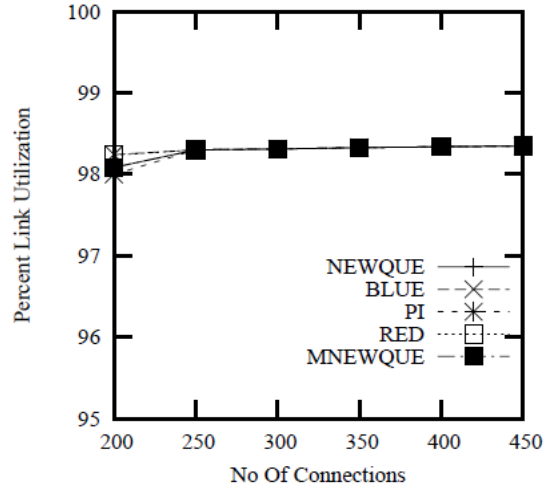


Figure 15 Percentage Link Utilization of Queue 3 w.r.to Number of TCP flows

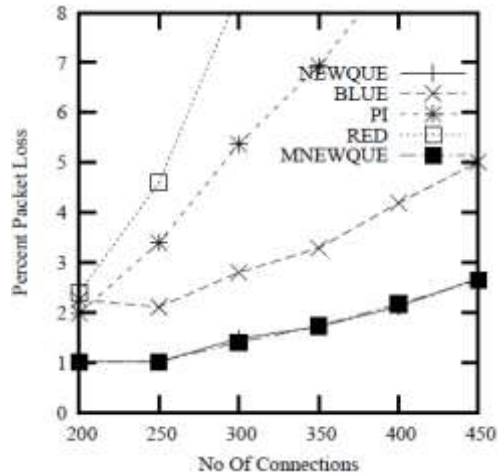


Figure 13 Percentage Packet Loss of Queue 3 w.r.to Number of TCP flows

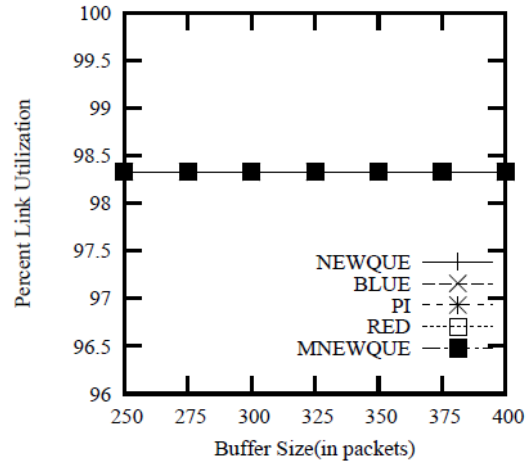


Figure 16 Percentage Link Utilization of Queue 3 w.r.to Buffer Size (in packets)

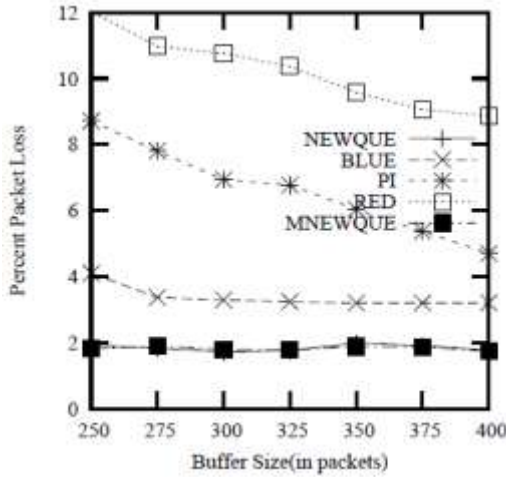


Figure 14 Percentage Packet Loss of Queue 3 w.r.to Buffer Size (in packets)

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

5. CONCLUSION

In this paper, we proposed a MNEWQUE AQM scheme supporting ECN. It is simple active queue management, improved than NEWQUE AQM. MNEWQUE is compared with four other AQM methods, BLUE, PI, RED and NEWQUE. The performance metrics used for the comparison are average queuing delay, link utilization and packet loss. The simulation experiments showed that the planned MNEWQUE AQM method performs better than BLUE, PI, RED and NEWQUE in terms of percentage packet loss and average queuing delay. It has same percentage link utilization than other AQM methods. 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.

6. REFERENCES

- [1] B. 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] 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.
- [5] S. Floyd, "Congestion Control Principles", RFC 2914, September 2000.
- [6] S. Floyd and V. Jacobson, "Random early detection for congestion avoidance", *IEEE/ACM Transactions on Networking*, Vol. 1, pp 397-413, July 1993.
- [7] V. Jacobson, "Congestion avoidance and control", in *Proc. ACM SIGCOMM*, Aug 1998, pp. 314-329.
- [8] The network simulator ns-2 (2000). [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [9] 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.
- [10] V. Santhi, Dr. A.M. Natarajan, "NEWQUE: A New Approach to Active Queue Management for TCP with ECN", in *International Journal of Computer Science and Network Security*, Vol.10 No.3, March 2010.
- [11] K. Ramakrishnan, S. Floyd and D. Black, "The addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, Sep. 2001.
- [12] 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.
- [13] 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.
- [14] 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.