IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN : 2278-0661, p-ISSN : 2278-8727
PP 12-17
www.iosrjournals.org

Effect of Router Buffer on TCP/UDP Performance

Ambhore P.G.[1], Mahajan S.G[2]

[1]*(Computer Engg. Department, JSPM's RSCOE, Pune India)*
[2]*(Computer Engg. Department, JSPM's RSCOE, Pune India)*

Abstract : *In data communication, congestion occurs when there are so many packets contending for the limited shared resources, such as the queue buffer in the router at bottleneck link. During congestion, large amount of packets experience delay or even be dropped due to the queue overflow. Severe congestion problems result in degradation of the throughput and large packet loss rate. In this paper, considering two popular queue management methods, Random Early Detection (RED) and LALRED the analysis of throughput and packet loss rate for different buffer sizes is presented. The results show that with increase in buffer size for RED, the throughput and efficiency of TCP increases. Also, UDP packet loss decreases with an increase in buffer size while the buffer size has no key impact on UDP packet delays. Simulation results obtained using NS2 establish the improved performance of LALRED over the RED and the graphs are plotted using GNU plot.*

Keywords *- TCP-Reno, NS2, UDP, Packet delay, Throughput analysis, Router buffer, Packet loss, Congestion window.*

I. INTRODUCTION

One of the main advantages that wired networks offer is their higher degrees of reliability and better connection strength as compared to their wireless counterparts. However, the performance of wired networks often degrades to a great extent due to congestion in the network. TCP is a transport layer protocol used by applications that require guaranteed delivery, mainly used in fixed networks. It is a sliding window protocol that provides handling for both timeouts and retransmissions. TCP establishes a full duplex virtual connection between two endpoints. Each endpoint is defined by an IP address and a TCP port number. The operation of TCP is implemented as a finite state machine. The byte stream is transferred in segments. The window size determines the number of bytes of data that can be sent before an acknowledgement from the receiver is necessary. UDP on the other hand is a connectionless and unreliable protocol. There is no such concept of windowing or retransmission. No packet has any knowledge of the preceding or following packet. The recipient does not acknowledge packets, so the sender does not know that the transmission was successful. UDP has no provisions for flow control; packets can be received faster than they can be used. In a packet switched network, end-to-end latency of individual packets is an important performance metric, which quantifies the behavior of the system from a user's point of view. End-to-end latency of a given packet consists of three main components.

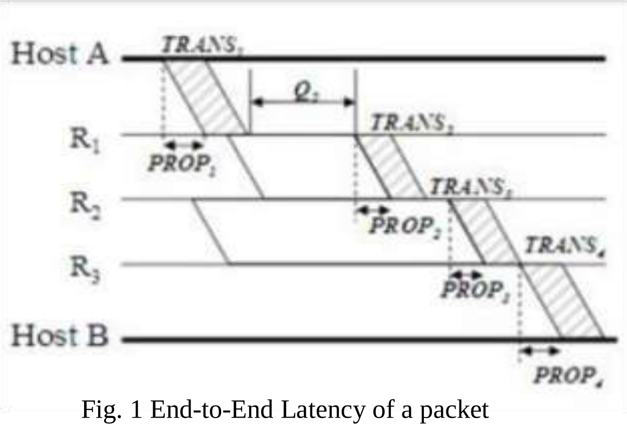


Fig. 1 End-to-End Latency of a packet

A. Transmission delay: This is the time it takes for a packet to be transmitted by the source host, and by any intermediate router on its path.

B. Propagation delay: This is the time it takes for a packet to reverse the links Connecting routers.
C. Queuing delay: This is the time that the packet sits in a buffer and waits for some system resource - usually the output port of the router which is blocked by other packets - to be released.
These three components, the first two (i.e. the transmission delay and the propagation delay) are fixed. Queuing delay is the only variable component of the end-to-end latency, and therefore it is what causes the variation in performance observed by the end users. In fact, one can argue that queuing delay is the single biggest cause of uncertainty in today's Internet as buffers cause queuing delay and delay variance, when they overflow they cause packet loss, and when they underflow they can degrade throughput Clearly, queuing delay and jitter are directly related to the buffer sizes. Today's Internet routers are set based on a rule- of - thumb which says, if we want the core routers to have 100%. Utilization, the buffer size should be greater than or equal to 2T ×C, also known as the bandwidth-delay product. Here, 2T is the effective round-trip propagation delay of a flow through the router (also denoted as RTT), and C is the capacity of the bottleneck link. In a recent paper, Appenzeller et al challenged this rule-of-thumb and showed that for a backbone network, the buffer size can be divided by $\surd N$ i.e. $B= 2T\times C/\surd N$ without sacrificing throughput, where N is the number of flows sharing the bottleneck [1]. The issue of router buffer sizing is still open and significant. In this study, we are analyzing throughput, packet delay, congestion window size and packet loss rate for different buffer size taking into account. AQM two popular queue management methods, Random Early Detection (RED) and LALRED, for different TCP and or UDP Poisson streams coming to a common router buffer with Exponential processing times.

II. METHODOLOGY

In this study, we are comparing throughput ,packet delay ,congestion window size (TCP Only) and packet loss rate for different buffer size taking into account AQM two popular queue management methods, Random Early Detection (RED) and LALRED, for different TCP (RENO) and UDP Poisson streams coming to a common router buffer with Exponential processing times. We will give the definition first.
A. Throughput: Throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot [2].
B. RED and LALRED
Random Early Detection (RED) [10] seeks to prevent the router's ueue from becoming fully used by randomly dropping packets and send signals to the sender to slow the sender down before the queue is entirely full. Two parameters govern RED's behavior, RED-min (the lower threshold) and RED-max (the higher threshold). A RED router maintains a notion of the length of the queue. RED routers maintain a running average of their queue length. When the queue length of some line exceeds a threshold, the line is said to be congested and action is taken. A temporary increase in the queue length notifies the transient congestion, while an increase in the computed average queue size reflects longer-lived congestion and RED router will send randomized feedbacks to some of the connections to decrease their congestion windows. The probability that a connection is notified of congestion is proportional to that connections share of the throughput through the RED router [11].

RED is a congestion-avoidance algorithm. RED is congestion-avoidance algorithm because RED foresees (or anticipates) the congestion by monitoring the average queue size. It also avoids global synchronization by randomly choosing packets to be marked or dropped before the queue gets full. The performance of RED is known to be sensitive to its parameters such as the Maximum threshold (MAX_{th}), the Minimum threshold (MIN_{th}), the Maximum packet-marking probability (PMP) (Max_P), and the so-called weighting factor [9]. Before we proceed,we clarify how these parameters affect RED. Let *Avg* denote the average queue size. So, following conditions are available in RED [8].

1) If $Avg < MIN_{th}$, then no packet drops and marks occur.
2) If $Avg < MINt_h$, then no packet drops and marks occur.
3) If $MIN_{th} < Avg < MAX_{th}$, then the packets are randomly marked with a certain probability whose value varies from zero to Max_P , evaluated using (2).
4) Let p_b be an intermediate PMP given by

$$p_b \leftarrow \max{}_p \times \frac{Avg - MIN_{th}}{MAX_{th} - MIN_{th}} \qquad (1)$$

Then, the Final PMP p_a is evaluated as per equation as follows:

$$p_a \leftarrow p_b \times \frac{p_b}{1 - Count \times p_b} \qquad (2)$$

Where *Count* denotes the number of the packets last marked

Before we proceed with our LAL-based solution, we have to clearly indicate the "Actions" which the environment has to offer (which the LAL scheme has to choose from). In our approach, we advocate the following four actions, based on the packet drop type.

- *Forced_Drop($Avg > MAX_{th}$.):* This action is chosen when the average queue size is above the maximum-threshold set for the queue or when the queue is full [24].
- *Minimum_Exceed:* This action is chosen when the average queue size exceeds the minimum threshold or it transitions from an empty queue state to a nonempty queue state.($MIN_{th} < Avg < Max_{th}$ and when *Avg* just crosses MIN_{th}.)
- *Unforced_Drop:* This action is chosen when the average queue size in between the minimum threshold and the maximum threshold**.** For an unforced drop, the arriving packet is always dropped. ($MIN_{th} < Avg < MAX_{th}$)
- *No_Drop*: This action is chosen when the average queue size lies below the minimum threshold.($Avg < MIN_{th}$)

The mutually exclusive nature of the actions is because the earlier four cases are themselves mutually exclusive. The rationale behind LALRED approach is as follows**.** First of all, we station a LAL machine, which makes its decisions based on a LALRED strategy. Fig. 2, shows Example of the transmission of packets from one network to another using a LAL machine placed at the gateway where we consider two networks: Network 1 and Network 2.

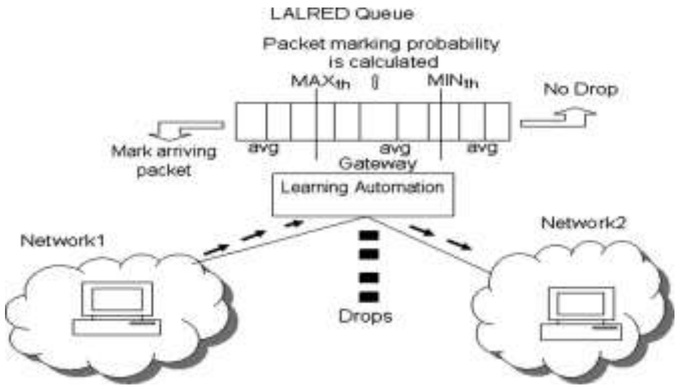


Fig.2 Example of the transmission of packets from one network to another using a LAL machine placed at the gateway

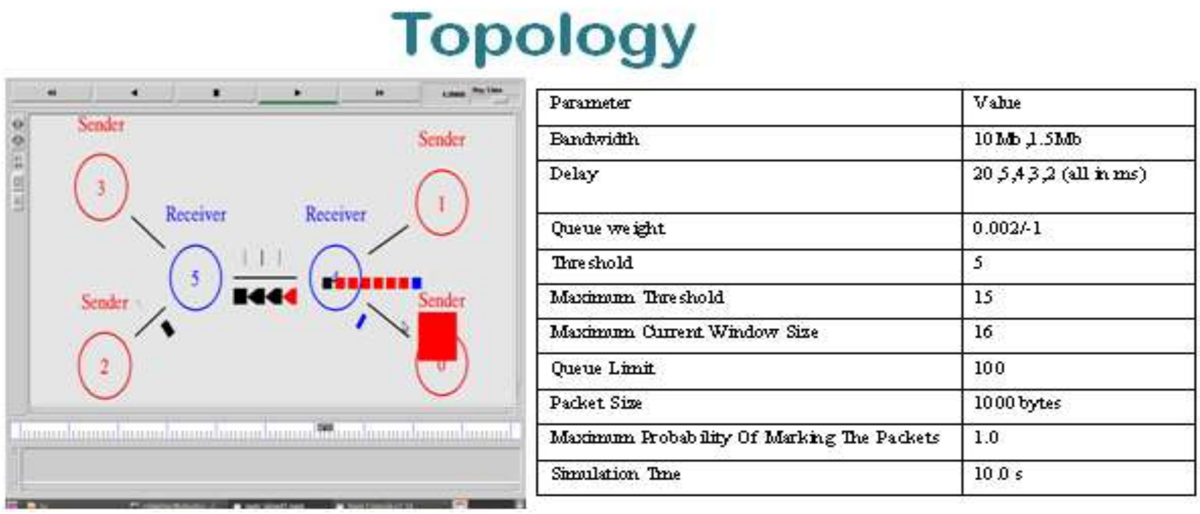
III. SIMULATION

We use the Network Simulator (NS2-2.37) [5, 6].The NS2 has all the essential features like abstraction, visualization, emulation, and traffic and scenario generation.

A Traffic Generation:

1) *CBR:* The CBR service category is used for connections that transport traffic at a constant bit rate.
2) *FTP:* Standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet.

B. Network Topology:



Topology

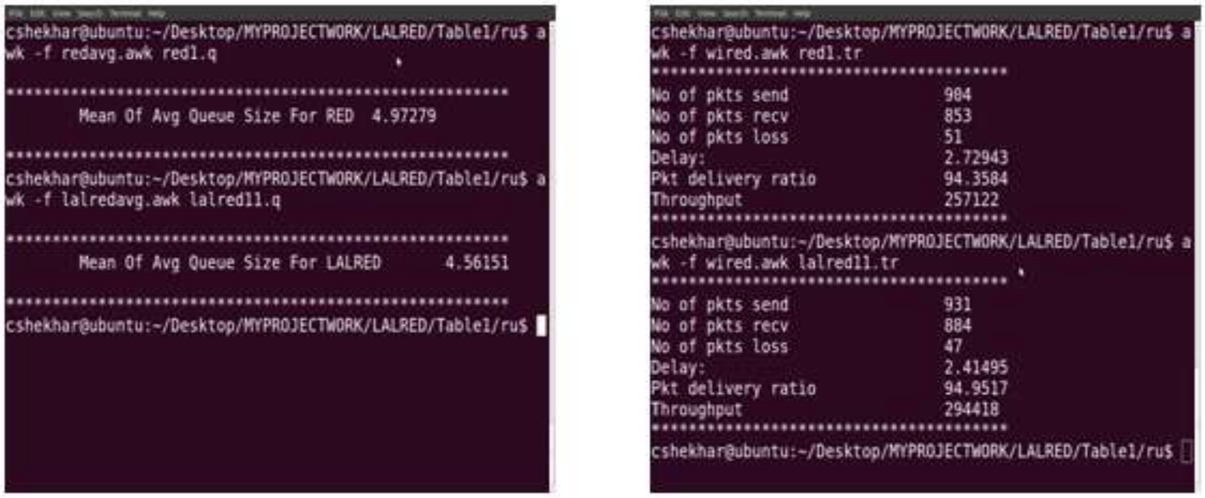
Parameter	Value
Bandwidth	10 Mb ,1.5Mb
Delay	20 ,5,4,3,2 (all in ms)
Queue weight	0.002/-1
Threshold	5
Maximum Threshold	15
Maximum Current Window Size	16
Queue Limit	100
Packet Size	1000 bytes
Maximum Probability Of Marking The Packets	1.0
Simulation Time	10.0 s

Fig.3 Topology of the network having six nodes

A. TCP-Throughput

TCP throughput has been analyzed with the buffer size impact on it. It has been found that with increasing buffer size the throughput and the performance (low-delay + less retransmission of packets) of the TCP increases.

This set of experiment aims at testing Packet loss, delay and throughput between RED and LALRED.Fig3 shows the network topology with network parameter which also clearly displays the senders and the receivers.



```
cshekhar@ubuntu:~/Desktop/MYPROJECTWORK/LALRED/Table1/ru$ a
wk -f redavg.awk red1.q

****************************************************

        Mean Of Avg Queue Size For RED  4.97279

****************************************************
cshekhar@ubuntu:~/Desktop/MYPROJECTWORK/LALRED/Table1/ru$ a
wk -f lalredavg.awk lalred11.q

****************************************************

        Mean Of Avg Queue Size For LALRED       4.56151

****************************************************
cshekhar@ubuntu:~/Desktop/MYPROJECTWORK/LALRED/Table1/ru$
```

```
cshekhar@ubuntu:~/Desktop/MYPROJECTWORK/LALRED/Table1/ru$ a
wk -f wired.awk red1.tr
**************************************
No of pkts send                 984
No of pkts recv                 853
No of pkts loss                 51
Delay:                          2.72943
Pkt delivery ratio              94.3584
Throughput                      257122
**************************************
cshekhar@ubuntu:~/Desktop/MYPROJECTWORK/LALRED/Table1/ru$ a
wk -f wired.awk lalred11.tr
**************************************
No of pkts send                 931
No of pkts recv                 884
No of pkts loss                 47
Delay:                          2.41495
Pkt delivery ratio              94.9517
Throughput                      294418
**************************************
cshekhar@ubuntu:~/Desktop/MYPROJECTWORK/LALRED/Table1/ru$
```

Fig 4: Comparison of RED and LALRED with average queue size, throughput, delay

Throughput is the average rate of successful message delivery .The delay of network specifies how long it takes data to travel across the network from one node or end point to another.Fig4 shows the comparison of RED and LALRED with Packet loss, delay and throughput and the mean of the average queue size of LALRED calculated for this plot is 4.56151, whereas that of RED is 4.97279. Fig5 shows graph for the average queue size. The instantaneous queue size for LALRED is also almost always less than that for RED. Fig 5 shows instantaneous queue size for LALRED is also almost always less than that for RED.

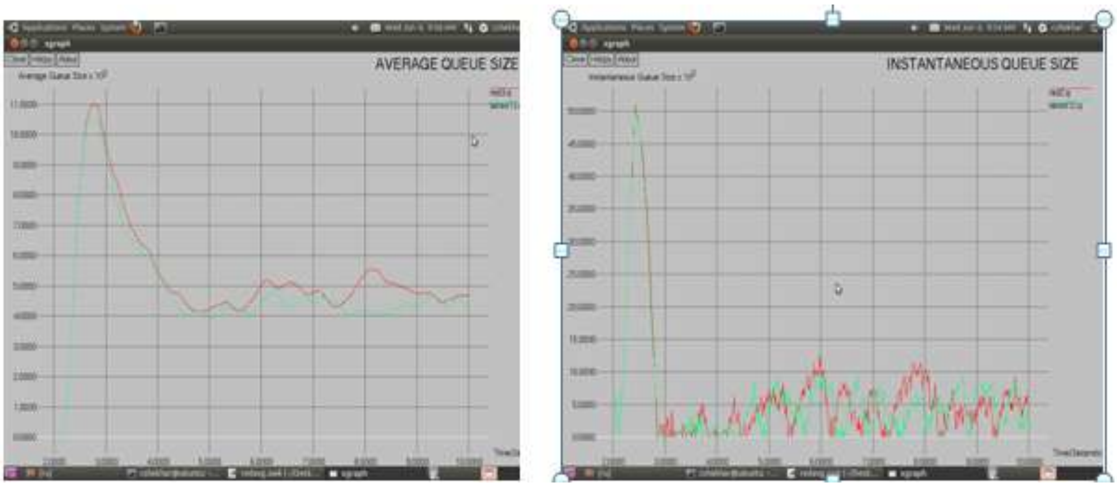
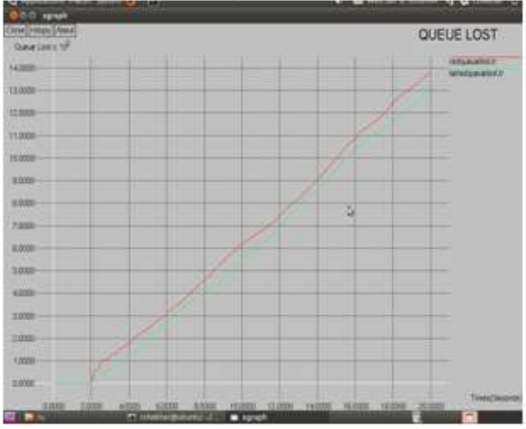


Fig 5:Average queue size and instantanious queue size



Parameter	Value
Bandwidth	10Mb
Delay	2 ms
Number of Nodes	6
Threshold	5
Maximum Threshold	15
Number of Flows	10
Queue Limit	25
Packet Size	1000 bytes
Maximum Probability Of Marking The Packets	0.5
Simulation Time	20.0 s
Queue type	Drop Tail or RED

Fig 6: Queue Lost with specified parameters

Fig.6 shows the comparison of queue lost of the RED and LALRED schemes. From the figure, it is shown that the curve of LALRED is almost always lower than that of RED. This signifies that the queue lost for RED is greater than that of LALRED.

IV. CONCLUSION

The study of router buffer sizing should not focus on TCP alone, but should consider the impact of real-time traffic also. Impact of buffer size increment/decrement has been analyzed for different transmission parameters like packet loss, packet delay and in case of TCP congestion window size etc. So the user can get a whole image of what is happening behind the scene as the choking throughput scenarios became very critical in certain situations. we examined the dynamics of UDP and TCP interaction at a bottleneck link router equipped with very small buffers. For LALRED algorithm the concept of a LAL mechanism devised for congestion avoidance in wired networks. LALRED uses the so-called estimate vector maintained by the family of Pursuit algorithms and updates the probability vector using a discredited philosophy so as to move toward convergence. LALRED is founded on the principles of the operations of existing RED congestion-avoidance mechanisms, augmented with a LAL philosophy, and it aims to optimize the value of the average size of the queue used for congestion avoidance and to consequently reduce the total loss of packets at the queue. Simulation results obtained using NS2 establish the improved performance of LALRED and the traditional RED methods which were chosen as the benchmarks for performance comparison purposes. From these, we infer the following results.

1) The number of packets lost at the gateway using LALRED is lower as compared to that using RED.
2) The average queue size maintained when using LALRED is lower as compared to that using RED.

LALRED should provide for congestion avoidance in both infrastructure based and infrastructure less wireless networks is one of the avenue for further research. As well the scalability of LALRED for use in networks having a large number of nodes needs to be improved. Finally, for future consideration we aim to perform simulations with various other versions of TCP such as TCP New-Reno, TCP Vegas etc., and emerging congestion control algorithms designed specifically for routers with very small buffers [7].

REFERENCES

[1]. Mihaela Enachescu, Yashar Ganjali †, Ashish Goel‡, Nick McKeown†. " Routers with Very Small Buffers"

[2]. Neeraj Bhargava, Dr. Ritu Bhargava, Bharat Kumar,Shilpi Gupta, Naresh Kumar Senwaliya, Kamal Kumar Jyotiyana "Implementation of TCP_Reno Algorithm in the ns2"

[3]. Bonald,T. etal." Analytic evaluation of RED Performance".Proceedings of the 19th Annual Joint Conference on the IEEE Computer and Communications Societies,Mar. 26-30, IEEE Xplore Press, USA., pp:1415-1424.

[4]. Floyd S., R. Gummdi and S. Shenker. "Adaptive RED: An algorithm for increasing the robustness of RED."2001".

[5]. NS - 2 Simulator tutorial ISI - documentation,[online] Available http:// www.isi.edu/nsnam/ns.

[6]. VINT DARPA-funded research project. [Online] Available: http://www.isi.edu/nsnam/vint.

[7]. Y.Gu, D. Towsley, C. V.Hallot and H.Zhang, "Congestion Control For Small Buffer High Speed Networks," Proc IEEE INFOCOM, Alaska, USA, May 2007

[8]. S.Misra, B.J.Oommen and S.Yanamandra, "Random Early Detection for Congestion Avoidance in Wired Networks: A Discretized Pursuit Learning-Automata-Like Solution" IEEE Trans.on system ,man,and cybernetics—part B: cybernetics, vol. 40, no. 1,February 2010

[9]. S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance," IEEE/ACM Trans. Networking, vol. 1, no. 4, pp.397–413, Aug. 1993.

[10]. M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, "Learning automata: Theory, paradigms, and applications," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 32, no. 6, pp. 706–709, Dec. 2002.

[11]. B. J. Oommen and M. Agache, "Continuous and discretized pursuit learning schemes: Various algorithms and their comparison," IEEETrans. Syst., Man, Cybern. B, Cybern., vol. 31, no. 3, pp. 277–287,Jun. 2001.

[12]. R. Fengyuan, R. Yong, and S. Xiuming, "Enhancement to RED algorithm,"in Proc. 9th IEEE Int. Conf. Netw., Oct. 2001, pp. 14–19.

[13]. http://ceit.aut.ac.ir/~bakhshis/ns-n%20Introduction%20to%20NS,%20Nam%20and%20OTcl%20scripting.pdf

[14]. www.roman10.net/how-to-add-a-new-aqm-protocol-in-ns2

[15]. http://ns-2.blogspot.in/2006/03/how-do-add-new-module-in-ns-2.html

[16]. Verma, A. Iyer, and A. Karandikar, "Active queue management using adaptive RED," J. Commun. Netw., vol. 5, no. 3, pp. 275–281,Sep. 2003.

[17]. B. J. Oommen and E. R. Hansen, "The asymptotic optimalityof discretized linear reward–inaction learning automata," IEEETrans. Syst., Man, Cybern., vol. SMC-14, no. 3, pp. 542–545,May/Jun. 1984.

[18]. http://en.wikipedia.org/wiki/active queue management.

[19]. B. Zheng and M. Atiquzzaman, "Low pass filter/over drop avoidance (LPF/ODA): An algorithm to improve the response time of RED gateways," Int. J. Commun. Syst., vol. 15, no. 10, pp. 899–906, Dec.2002.

[20]. NS2 Network Simulator, Last accessed on Feb. 3, 2009. [Online] Available: http://www.isi.edu/nsnam/ns/

[21]. http://www.cs.virginia.edu/~cs757/slidespdf/cs757-ns2-tutorial1.pd. S. Narendra and M. A. L. Thathachar, Learning Automata: An Introduction. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[22]. Active Queue Management, Last accessed on Feb. 3, 2009. [Online].Available: http://en.wikipedia.org/wiki/active queue management