Fuzzy Adaptive Tuning of Router Buffers for Congestion Control

S. Mercy Shalinie, G. Preetha, S. Dina Nidhya, B.S. Kiruthika Devi
Department of Computer Science and Engineering,
Thiagaraja College of Engineering,
Anna University, India
Email id : kiruthi_23@yahoo.com

Abstract

Network Congestion occurs when the aggregate demand for bandwidth exceeds the available capacity of a link and when the arrival rate to the router is greater than its departure. To solve this problem, it is necessary that the router must implement effective queuing algorithm that governs how packets are buffered while waiting to be transmitted. According to the dynamic of input packets and available link bandwidth, queue management becomes very complex. For this reason it is better to use an intelligent algorithm. In this paper, a Fuzzy Inference System implementation for Drop Tail, Adaptive Drop Tail-Fuzzy Logic (ADT-FL) is proposed which regulates the queue size of the router buffers based on prevailing traffic conditions and available link bandwidth preventing the router buffers from becoming full when congestion occurs. Simulation results are provided to show that ADT-FL has better performance and lower loss rate than the conventional queuing algorithms used in current internet buffers.

Keywords: Fuzzy Inference System, Drop Tail, Queue management, Router buffers

1. Introduction

Congestion occurs in shared networks when multiple users contend for access to the same resource (bandwidth, buffers, and queues). Internet routers use queues to smooth spikes in incoming packet rates. Each router uses two classes of algorithms, namely queue management and scheduling. Queue management algorithms manage the buffer at the router by determining which packets to drop or mark for congestion notification. Scheduling algorithms determine which packet to forward and manage the allocation of bandwidth among the incoming flows.

When the router's incoming packet rate exceeds the departure rate, then the available queue size starts increasing as time increases. A full queue condition is achieved where the new incoming packet finds the queue full and finally it is dropped due to lack of buffer space. The most popular Drop Tail policy [9] drops the incoming packets once the buffer is full. With this policy packets are enqueued at the tail of a queue as they arrive and dequeued from the head of queue when there is capacity on the link. The arriving packets are dropped if the queue is full. Moreover, Drop Tail policy suffers from lockout and full queues. Drop Tail with heavy traffic loads leads to high packet loss rate, large queuing delays, low throughput and long lasting congestion [6][3].

To solve the above mentioned problem, Active Queue Management (AQM) techniques [1] have been proposed. It tries to detect and react before buffer overflow occurs. Active queue management algorithms drops packet actively to notify the traffic sources to slow down its transmission rate.

RED is one of the first proposed solutions for active queue management. RED is a modification to Drop Tail which sets a minimum(min_{th}) and maximum threshold(max_{th}).The average queue size is calculated every time a new packet arrives. If the average queue size exceeds the minimum threshold, RED starts dropping packets based on a probability depending on average queue size. If the average queue size exceeds the maximum threshold, then every packet is dropped. RED uses randomization to solve both the lockout and full queues. RED uses queue length as congestion indicator. But it varies with the level of congestion and parameter settings. As a result, RED performance is too sensitive to parameter setting and traffic load [7]. Though AQM variant RED is claimed to be a better solution than Drop tail it fails for different network conditions due to selection of congestion indicators.

A need for an efficient algorithm arises in complex networks which takes input traffic rate and queue occupancy with reduced packet loss. Internet routers are facing packet loss and queuing delays as the amount of data carried increases. This includes the voice over IP and video streams. The necessity for an efficient congestion control and queue management algorithm comes into play.

The fuzzy logic controller finds to be a remarkable solution for congestion control in networks since a precise mathematical model using conventional analytical methods is very difficult to be obtained [8]. Fuzzy logic controllers are expected to work in situations where there is a large uncertainty or unknown variation in the parameters and structures of the system under control.

The most important advantage of adaptive fuzzy logic controllers over conventional adaptive control is that adaptive fuzzy logic controllers are capable of incorporating linguistic information from human operators or experts [4], where conventional controllers cannot. This attribute is important for systems with a high degree of uncertainty. Adaptive fuzzy logic control provides a tool for making use of the fuzzy information in a systematic and efficient way.

2. The Proposed Fuzzy Inference System

2.1. Adaptive Drop Tail Using Fuzzy Logic (ADT-FL) Controller

In this section, a fuzzy logic approach for congestion control, ADT-FL is proposed which is an extension to the traditional Drop Tail mechanism. The traditional drop-tail queue maintains FIFO queue, implemented by including an object of the *PacketQueue* class in ns-2 [10]. Drop-tail implements *enque* and *deque* as follows:

```
void DropTail::enque(Packet* p)
{
      q_.enque(p);
      if (q_.length() \>= qlim_)
     {
            q_.remove(p);
            drop(p);
      }
}
Packet* DropTail::deque()
{
      return (q_.deque());
}
```

Here, the *enque* function first stores the packet in the internal packet queue and then checks the size of the packet queue versus *qlim_*. With tail drop, when the queue is filled to its maximum capacity, the newly arriving packets are dropped until the queue has enough room to accept incoming traffic inducing high packet loss. FIFO scheduling is implemented in the *deque* function by returning the first packet in the packet queue.

The Fuzzy-ADT [5] controls the queue size *qlim_* of the Drop Tail algorithm under varying traffic intensities and link bandwidth in order to save the router from congestion, subsequently preventing the network from congestion.

The main part of designing the Fuzzy ADT-FL is to design the ADT-FL Controller which acts as a congestion controller in the routers. Figure 1 shows the design structure of the proposed FLC. The proposed model consists of two input Traffic Intensity and Available Link Bandwidth and Queue size of the buffer is controlled at the output. The basic elements of FLC are the fuzzifier, fuzzy inference, fuzzy rule base, and a defuzzifier. The input and output of this system are described by fuzzy sets. In this proposed model, there are two inputs, Traffic Intensity and Available Link Bandwidth which are denoted by Trafint and Av_{bw} respectively. Output of this model is Queue size parameter denoted as $Q_{size.}$

The linguistic rules are defined as follows:

If $Traf_{int}$ is A_1 and Av_{bw} is B_1 then Q_{size} is C_1
If $Traf_{int}$ is A_2 and Av_{bw} is B_2 then Q_{size} is C_2
If $Traf_{int}$ is A_3 and Av_{bw} is B_3 then Q_{size} is C_3
……………..   .
If $Traf_{int}$ is A_k and Av_{bw} is B_k then Q_{size} is C_k

In =

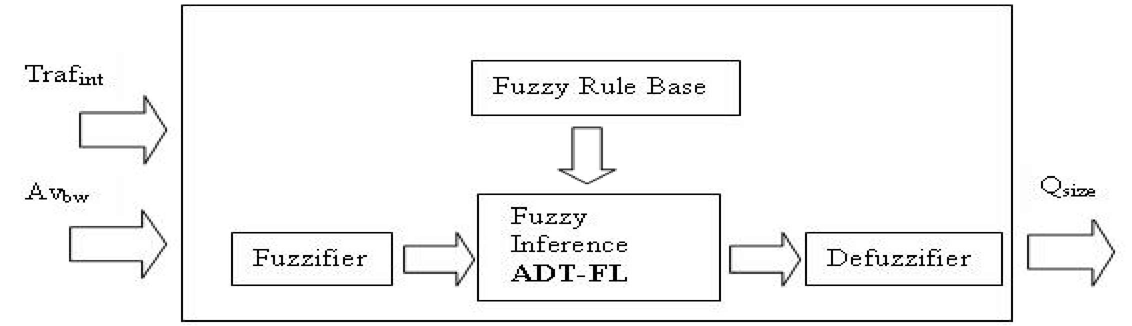

Fig.1: Structure of Fuzzy Logic Controller

In this paper, the proposed Adaptive Drop tail Fuzzy Logic (ADT-FL) calculates Q_{size} parameter based on the traffic Intensity, available bandwidth with a set of fuzzy rules. In order to simplify the computational process triangular membership function is used. The triangular membership function is described as in equation (1).

$$\mu(x;\, a, b, c) = \left.\begin{cases} 0 & , \text{for } x > 0 \text{ or } x > c \\ \dfrac{x - a}{b - a} & , \text{for } a \le x \le b \\ \dfrac{c - x}{c - b} & , \text{for } b \le x \le c \end{cases}\right\} \text{———} \quad (1)$$

2.2 Fuzzy Sets, Membership Functions and Rules

In order to calculate ADT-FL Q_{size} parameter, the definition of fuzzy sets, membership functions and number of rules for the inference system are needed. The Traffic Intensity and Available Link Bandwidth Trafint and Av_{bw} are classified into three linguistic variables: {Low, Medium, High}. The input membership functions are as shown in Fig. 2.

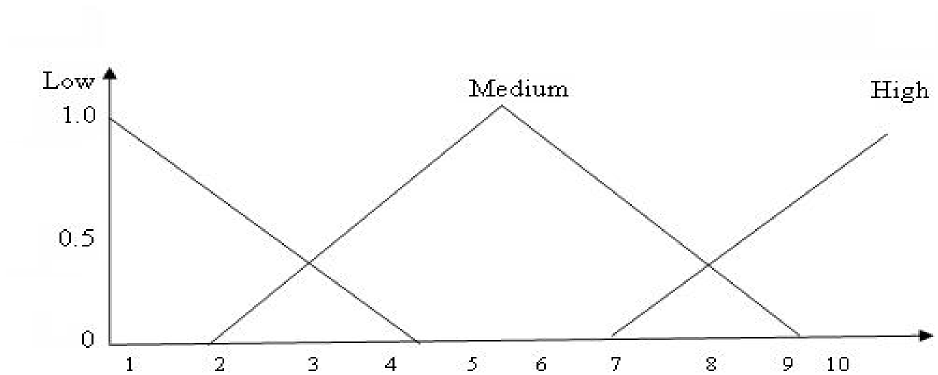

Figure 2: Input membership function- Traffic Intensity & Available link bandwidth

Output function of Q_{size} is classified into five linguistic variables: Q_{size} = {Very Low, Low, Medium, High, Very High} which simplified as Q_{size} = {VL, L,M, H, VH}. Figure3 shows the Q_{size} output membership function. When a packet arrives inside the router interface buffer, the current value of $Traf_{int}$ and Av_{bw} are obtained and Q_{size} value is calculated based on two inputs and set of rules defined on Table1.

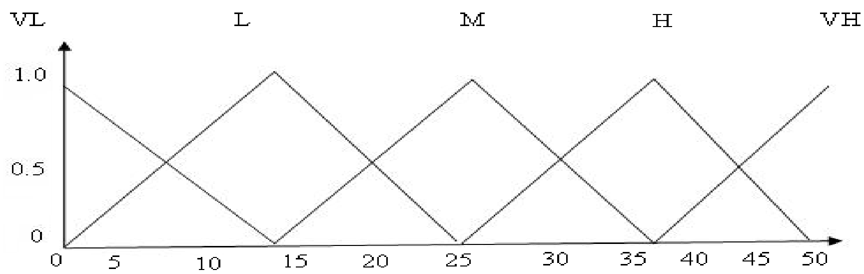

Figure 3: Output membership function- Queue Size

Table1: Fuzzy Rules for Adaptive Drop Tail using Fuzzy Logic

Traffic Intensity(Mb)	Available Link bandwidth(Mb)		
	Low	Medium	High
Low	L	L	VL
Medium	M	M	L
High	VH	H	M

3. Simulation

The performance of the proposed fuzzy controller is compared with the traditional Drop Tail and RED mechanism. The simulation is done with the ns-2 network simulator [10] version 2.31 under Linux operating system.

In order to show the performance of the proposed fuzzy controller against traditional RED and Drop tail mechanisms, two simulation scenarios are used. The simulation scenarios show how each algorithm behave in maintaining the queue size when congestion level changes in a periodic time interval.

In the first scenario, the simple network topology with a single TCP and UDP source is used as shown in Figure 4 .The traffic agents are FTP and CBR over TCP [2] and UDP sources respectively. RED queue on ns-2 is set to “drop”, not to mark packets. The minimum and maximum thresholds for RED are set to 10 and 40 packets respectively. The buffer size for the traditional algorithms was set to 5 packets and the simulation lasted for 5 seconds. The traffic

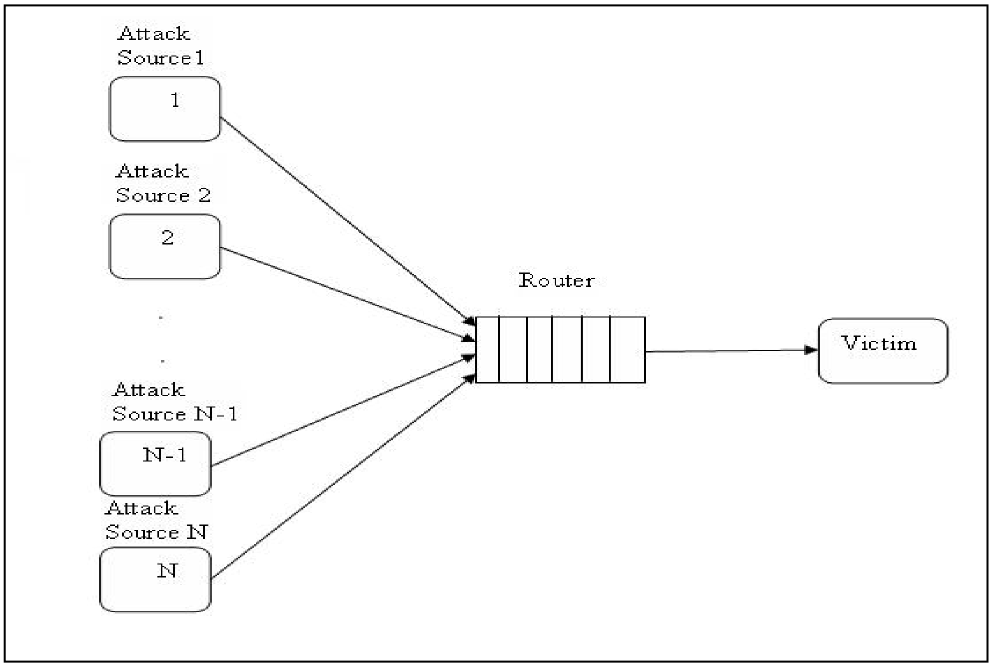

Figure 4: Network Topology in ns-2 simulation

intensity rate is varied from 0-0.5Mb and the link utilization is also varied from 0-0.5 Mb over the periodic time interval.

Packet Loss is calculated under varying traffic conditions .The packet loss for traditional Drop Tail, traditional RED and Adaptive Drop Tail using fuzzy Logic is compared and analyzed. The results for the number of packets lost for varying traffic intensity and link bandwidth using ns-2 simulation are shown in Figures 5 and 6 respectively. From the results, it is clear that as the traffic intensity increases, the number of packets lost increases with congestion [2] but not drastically as in case of traditional algorithms.

Due to changes in network traffic load, mix of traffic, mix of congestion control actions, on/off flows the statistics of arriving traffic is not stationary. The available link bandwidth varies in accordance with the statistics of the input traffic. ADT-FL logic adapts itself according to the varying traffic intensities and available link bandwidth and performs better than traditional Drop Tail and RED algorithms.

In the second scenario, a total of 13 sources comprising UDP and TCP flows with CBR and FTP traffic were used as shown in Figure 4. Simulation time was set to 5 seconds. Two sources were started at time 0. After 2 seconds eleven additional flows were started to simulate increase in congestion. After 4 seconds, the numbers of flows were returned to three sources.

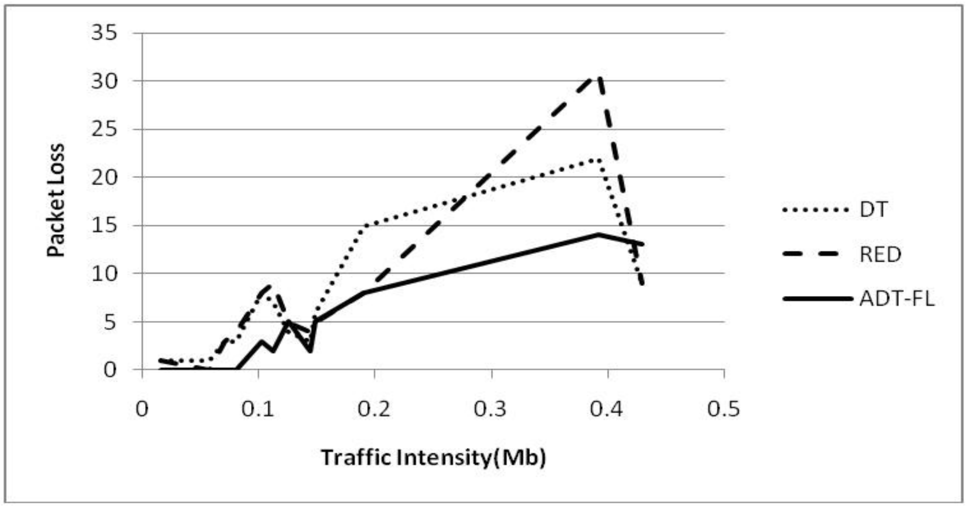

Figure 5: Comparison of Drop Tail, RED and Adaptive Drop tail-Using Fuzzy logic under varying Traffic intensity

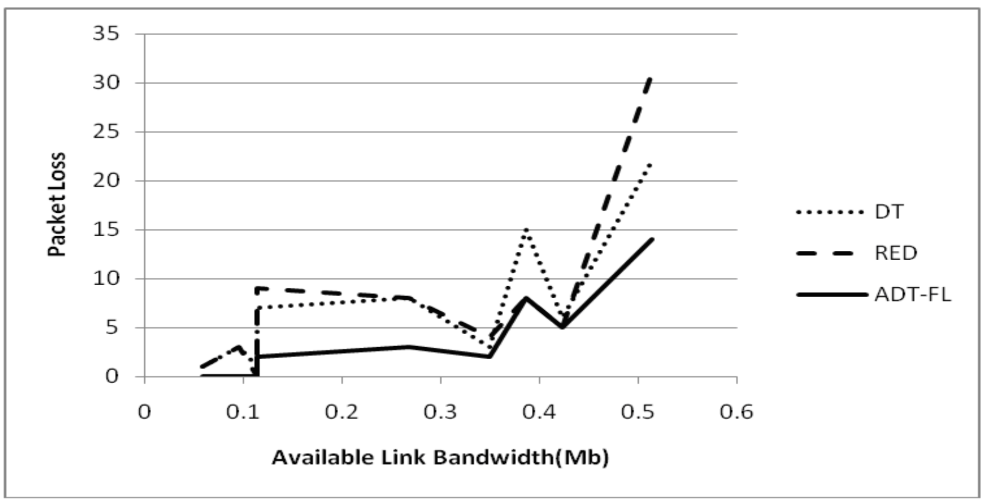

Figure 6: Comparison of Drop Tail, RED and Adaptive Drop tail-Using Fuzzy logic under varying Link Bandwidth

The simulation was done under varying rate of traffic intensity and link bandwidth. From the trace file generated using ns-2, the number of packets sent and lost is calculated using awk scripts. The results with varying queue sizes are shown in Table2. From the results, it is seen that the number of packets lost when using Fuzzy logic is very low when compared to traditional algorithms. Congestion occurs when the traffic intensity is high which implies that the packet arrival rate is high at the router. Adaptive Drop Tail using Fuzzy Logic can control the queue by adjusting it to a desired size and maintain an adequate level of packets in the queue under varying traffic and network conditions so that the packet loss remains at a low rate. ADT-FL behaves well with low packet loss when the queue size of the buffer is low. This in turn leads to high throughput, low queuing delay and jitter.

Table2: Packet Loss for DT, RED and ADT-FL under varying traffic conditions (Q_{size}= 5pkts)

Traffic Intensity(Mb)	Available Bandwidth(Mb)	Packets lost Conventional DT		Packets lost Conventional RED		Packets lost ADT-FL	
		Packets Lost	Loss%	Packets Lost	Loss%	Packets Lost	Loss%
1.45	9.63	172	8.95	103	5.12	174	8.89
2.36	8.24	514	22.91	397	17.14	0	0
2.53	7.75	292	10.52	165	5.97	0	0
3.77	4.07	780	23.07	731	21.58	582	17.74
3.52	5	777	25.61	642	20.03	232	7
5.5	7.5	1158	24.35	890	19.15	309	6.19
8.39	7.02	1776	26.39	1679	25.27	1564	23.86
8.8	5.7	2568	37.06	2619	37.45	2374	34.98
9.88	5	3419	44.81	3363	44.35	3295	43.61

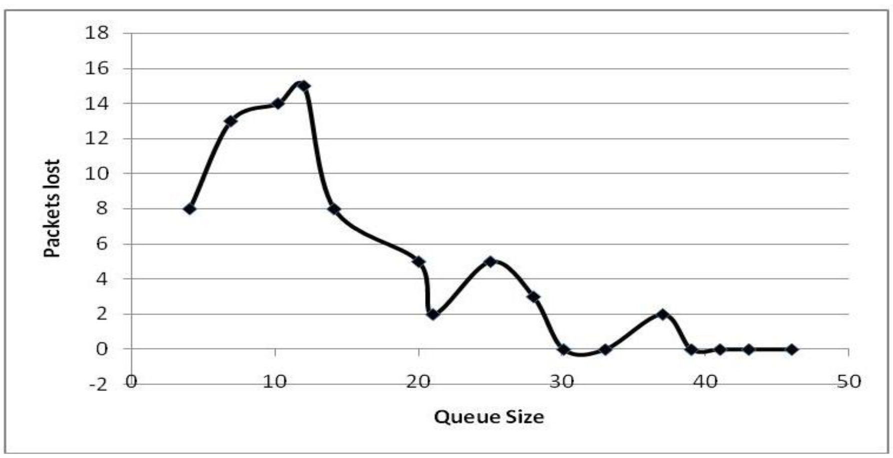

Figure 7: Buffer Queue Size Vs Number of packets Lost

The proposed fuzzy controller can maintain adaptive buffer with varying queue size and thus prevent the buffer from empty queue and full queue which is really a critical issue in current internet buffers. Figure 7 shows how ADT-FL logic adapts the available buffer space for changing traffic conditions thus minimizing the packet loss

4. Conclusions

In this paper, a fuzzy-based controlling mechanism is presented that is compatible with changing network and traffic conditions. When the incoming traffic is bursty, the current internet router buffers fail to control congestion effectively. But fuzzy logic based Adaptive Drop Tail shows significant improvement in controlling congestion without any need for special parameterization or tuning. Two experiments were done using ns-2 network simulator and the outcome shows that the Adaptive Drop Tail Fuzzy Logic controller has reduced packet loss when compared to traditional Drop Tail and RED mechanisms. The simulation is designed to maintain adaptive buffer space under a sudden change in congestion level which prevents the internet router buffers from becoming full when congestion occurs.

5. Acknowledgement

We thank the Smart and Secure Environment (SSE) project led by IITM, Chennai for providing excellent infrastructure**.** This work was supported in part by a grant from the National Technical Research Organization, New Delhi.

References

[1] G.Thiruchelvi and J.Raja, “A Survey on Active Queue Management Mechanisms”, IJCSNS International Journal of Computer Science and Network Security, Vol.8, No.12, December 2008, **pp.130-145**.

[2] Ibrahiem M. M. El Emary and Farouq Al Taweel, “An Advance Approach to Evaluate the Performance of the TCP Networks”, American Journal of Applied Sciences, Vol-2, Issue 9, 2005, pp.1375-1379.

[3] Monika Sachdeva, Krishnan Kumar, Gurvinder Singh and Kuldip Singh, “Performance Analysis of Web Service under DDoS Attacks”, IEEE International Advance Computing Conference (IACC 2009), March 2009, Patiala, India, pp. 1002-1007.

[4] Behrouz Safaiezadeh, Amir Masoud Rahmani and Ebrahim Mahdipour, “A New Fuzzy Congestion Control Algorithm in Computer Networks”, IEEE International Conference on Future Computer and Communication”, April 03-05, 2009, Kuala Lumpur, Malaysia, pp. 314-317.

[5] Rade Stanojevic, Robert N. Shorten and Christopher M.Kellett, “Adaptive Tuning of Drop-Tail Buffers for Reducing Queuing Delays”, IEEE Communication Letters, Vol. 10, Issue: 7, July 2006, pp. 570-572.

[6] Dharm Singh, Chih-Heng Ke, Gaurav Jain and Hari Narayan Sanadhya, “Measurement of Wireless Network Performance”, IEEE National Conference on Advanced Technologies and Applications, GITS, Udaipur, India, 13-14 May 2009.

[7] Tonny Adhi Sabastian, M. Rahmat Widyanto and Husni Fahmi, “RED Queue Improvement Via Maxp Tuning With Queue Ratio Fuzzy Inference System”, In Proceeding on The 2nd International Conference of IT Application & Management (ITAM) 2009, Universitas Indonesia, June 25th 2009.

[8] M.H. Yaghmaee and Halleh AminToosi, “A Fuzzy Based Active Queue Management Algorithm”, in the Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS2003), pp. 458-462, 20-24 July, Montreal, Canada, 2003.

[9] G. Iannaccone, C. Brandauer, T. Ziegler, C. Diot, S. Fdida and M. May, “Comparison of tail drop and active queue management performance for bulk-data and web-like internet traffic," in the Proceedings of ISCC (2001), IEEE, pp. 122-129.

[10] The Network Simulator – ns-2 homepage, available at http://www.isi.edu/nsnam/ns.