# Enhanced Random Early Detection (ENRED)

Alshimaa H. Ismail
Electronic & comm. Dept.,
Delta Higher Institute for Eng.& Tech., Delta Academy for science & Tec., Mansoura, Egypt.

Ayman EL-Sayed & Zeiad Elsaghir
Computer Science & Eng. Dept.,
Faculty of Electronic Eng.
Menouf 32952, Egypt.

Ibrahim Z. Morsi
Electrical Eng. Dept.
Faculty of Engineering,
Menoufia University, Egypt.

## ABSTRACT
As the internet is expected to better support many applications such as multimedia with limit bandwidth, new mechanisms are needed to control the congestion in the network. Active Queue Management (AQM) algorithms play an important role to ensure the stability of the Internet. Random Early Detection (RED) is the first active queue management algorithm proposed for deployment in TCP/IP networks. RED has some parameters tuning issues that need to be carefully addressed for it to give good performance under different network scenarios.We propose a new algorithm called Enhanced Random Early Detection (ENRED). ENRED works to improve these parameters to provide better congestion control over the network while remaining the advantage of RED.This paper will introduce ENRED and some features about RED and its variants**.** We simulate the proposed algorithm (ENRED) using the well-known network simulator ns-2, by comparing it to the original RED. Simulation results show that the proposed algorithm achieves better queue size than RED and decreases the delay and losses.

## Keywords
Active Queue Management, Congestion control, Queue size, RED, TCP/IP.

## 1. INTRODUCTION
Computer network have experienced an explosive growth over the past few years, that growth cause congestion collapse. When this congestion occurs performance will degrade. Congestion control mechanism is carried out in the transport layer [1, 2].

Transmission control protocol (TCP) is the most popular transport layer protocol for the internet. Due to various reasons, such as multipath routing, route fluttering, and retransmissions, packets belonging to the same flow may arrive out of order destination. Such packet reordering violates the design principles of some traffic control mechanism in TCP and, thus, poses performance problems [1].

TCP is a connection oriented reliable protocol. TCP is end-to-end congestion control where all the work is done by transport layer. It is extensively used in the internet, TCP uses a number of mechanisms to achieve high performance and avoid congestion collapse [1, 3, 4, 5]

In [1, 6], the author studies the stability issue of the average queue length of Transmission Control Protocol (TCP) model when interacting with Random Early Detection (RED). The model used for study has shown period doubling bifurcation (PDB) in the average queue size at certain values of parameters when original RED is deployed. They adopt a gentle version of RED and new derived RED algorithm into the model to study the improvement in stability of average queue size in the system.

In [1, 7], the authors analyze the dynamic behavior of a single RED controlled queue interacting with a large population of idealized TCP sources. The aggregate traffic from this population is modeled in terms of the time dependent expected value of the packet arrival rate which reacts to the packet loss tacking place in the queue. The queue is described in terms of the time dependent expected values of the instantaneous queue length and exponentially average queue length.

TCP congestion control has been designed to ensure internet stability along with fair and efficient allocation of the network bandwidth. Congestion control defines the methods for implicitly interpreting signals from the network in order for a sender to adjust its rate of transmission to prevent a sender from overrunning the capacity of the network [1, 8].congestion control is built as distributed mechanisms that prevent congestion before happen or even remove the congestion if it happened [9].

The main objective of congestion control mechanisms is to keep the network running pretty close to its rated capacity, even when faced with extreme overload. These objectives could be translated into two main goals, the first is to avoid the occurrence of network congestion before happen and dissolve the congestion if the congestion occurrence cannot be avoided. The second is to provide a fair service to the different connection, along with support various internet application domains with diverse Quality of Service (QOS) requirements [1, 10].

Generally, there are two ways to implement congestion control: (1) Network assistant congestion controls; they are approach taken by routers [11, 12]. These approaches use the router queue size to monitor the congestion state of the network. (2) End-to-End congestion controls; they are approaches taken by the transmission control protocol (TCP) and are mostly achieved in transport layer [12].

Active Queue Management (AQM) [13] routers have been recently proposed to support the End-to-End congestion control in the internet. AQM is an active approach than Droptail the earliest researcher trends to AQM as a solution to overcome the drawbacks of droptail technique which let packets drop if the queue is shorter than the packet maximum size and concerned with the problem of global synchronization which keep all senders stop transmission at the same time and retransmission at the same time [14]. AQM has been recommended by the Internet Engineering Task Force (IETF). It is a technique based on the following routine. The router queue first works on best effort service by marking or dropping the packet before the queue is full and also works on avoiding global synchronization [15].

The AQM algorithms can be classified according to the criteria on which the decision whether to drop packets from the queue or not. It has different issues in this can be

identified. First average queue length-based queue management (QM), Second packet loss & link utilization-based QM, Third class-based QM, fourth control theory-based QM [16, 17]. Where algorithms can be classified as being either reactive or proactive, a reactive AQM algorithm focuses on congestion avoidance, Congestion can occur in this case, but it will be detected early and alleviate. Decisions on actions to be taken are based on current congestion. A proactive AQM algorithm focuses on congestion prevention which known as open loop congestion control. In this case it is works to prevent congestion before it happens [15, 16, 17].

This paper is organized as follow: Section 2 describes the Transmission Control Protocol (TCP). The RED algorithm and Evaluation of its variants are described in section 3. Our proposed algorithm is described in Section 4.The performance evaluation is shown in Section 5. Finally the paper is concluded in section 6.

## 2. TRANSMISSION CONTROL PROTOCOL (TCP)

Transmission Control Protocol (TCP) [18, 19,20] is a reliable, connection oriented, end-to end, error free in order protocol. A TCP connection is a virtual circuit between two computers, conceptually very much like a telephone connection but with reliable data transmission between them. A sending host divides the data stream into segments. Each segment is labeled with an explicit sequence number to guarantee ordering and reliability. When the host receives in the segments sequence, it sends a cumulative acknowledge (ACK) in return, notifying the sender that all of the preceding segment's had been received. If an out-of-sequence segment is received, the receiver sends an acknowledgement indicating the sequence number of the segment that was expected. If outstanding data is not acknowledged for a period of time, the sender will timeout and retransmits the unacknowledged segments.

## 3. RANDOM EARLYDETECTION (RED)

RED is the basic of the reactive class of AQM algorithms, where it is maintains the features of this class in achieve the fairness between active data flow from available bandwidth, and it is an average queue length-based algorithms. RED algorithm takes its decision on whether or not to drop packets from the queue on the observed average queue length [15, 16, and 17]. RED is tailored for TCP connection across IP routers it's designed to avoid congestion, global synchronization, and avoidance of bias against traffic and bound on average queue length to limit delay. RED is a queue length that marks packets with probability proportional to the current average queue length, for each arriving packet the average queue size is calculated [14, 21, 23]. The average queue size is compared with the minimum threshold and maximum threshold to take its decision, at each arriving packet if average queue size is less than minimum threshold packet is in queue but if average queue size is larger than maximum threshold packet marked. And if the average queue size is in between minimum and maximum threshold packet is marked with probability where it is a function of measured queue length [14, 24]. So RED operation depends on calculating the average queue size using the Exponential Weight Moving Average (EWMA) [16, 21, 22, 25], and calculating the packet-marking probability. Calculation of $q_{avg}$ is carried out using the equation 1.

$$q_{avg} = (1 - w_q)q_{avg} + w_q \cdot q \text{---------- (1)}$$

where q is the instantaneous queue length as observed at the router, and $w_q$ is the weightapplied by the low-pass filter to the old average queue size.

The initial packetmarking probability $p_b$ is calculated as a linear function of the average queue size. It has two major methods to calculate the final packet-marking probability. The first method, when the average queue size is constant the number of arriving packets between marked packets is a geometric random variable. The second method the number of arriving packets between marked packets is a uniform random variable[25]. After $q_{avg}$has been calculated, it is compared to two threshold values, ***MINth*** and ***MAXth***. Then, the initial packet-marking probability is computed as shown in the equation 2.

$$p_b = P_{max} \times (q_{avg} - MINth)/(MAXth - MINth) \text{------ (2)}$$

Where $P_{max}$ is the maximum value for the probability of dropping packets $p_b$, achieved when the average queue size reaches the maximum threshold (***MAXth***).

RED has its own variants which tend to control average queuing delay, while still maintaining high link utilization, reducing packet drops, reducing global synchronization and burst connection [21, 25]. RED variants are an Implementing schemes, So that packets are transmitted with higher priority than others. All variants depend on RED parameters in dealing with congestion and achieving the highest quality of service QoS for router queue such as Adaptive Random Early Detection (ARED), D Stabilized Random Early Detection (SRED), Flow Random Early Detection (FRED), Dynamic Random Early Detection (DRED), Modified Random Early Detection (ModRED), all these algorithms are described in the following subsections.

### 3.1 Adaptive RED (ARED)

ARED minimizes both packet loss rate and the difference in queuing delay by keeping the average queue size doesn't exceed the half way between minimum threshold and maximum threshold [21, 26], ARED also work on not go underneath a packet loss probability of 1%, and it should not exceed a packet loss probability of 50% [27].

### 3.2 Stabilized RED (SRED)

SRED works to stabilize the queue size at a level independent of the number of active flows, The drop probability of packets is computed by obtaining the active flow to adjust instant queue size [27, 28].SRED maintain its own virtual cache which like a container called a zombie list. The zombie list stores both source and destination addresses for each arriving packet. When the zombie gets its full rate a random zombie is ejected from the list and compared with the source and destination addresses for the new packet. If it belongs to the same flow is set to one. Otherwise, it is set to zero, and with a certain probability P, the content of virtual list may be replaced by the source and destination of this new packet [14, 26].

### 3.3 Flow RED (FRED)

FRED is interested in keeping the state flow information. So it penalizes non adaptive connection by imposing a maximum number of buffered packets; FRED depends on calculating the average queue size at both packet arrival and departure [14, 29]. FRED is protecting week flows by accepting packets from low bandwidth flows. FRED provides fair sharing for large numbers of connection by accepting two- packet-buffer [16, 30].

FRED obtains two parameters (***MINq***) and (***MAXq***) which refer to both minimum and maximum number of packets gets from each flow to the buffer. FRED computes the average per-active-flow buffer usage by maintaining a global variable (***avgcq***). It maintains the number of active flow and when the flow is active (***Qlen***>***MAXq***) it maintains a count of buffer packets (***Qlen***), and a count of times [22, 29].

## 3.4 Dynamic RED (DRED)

DRED aims to minimize the queue size. DRED is controlling the packet loss rate in a simple feedback control approach to discard packets randomly [14, 26]. The packet marking probability of DRED is responsive enough to traffic and as a result the buffer will not overflow at the gateway. The DRED operation is identified as a sequence of steps, carried out at time n. First, the instant queue size ***Q(n)*** is sampled. Second, the instant error signal ***E(n)*** is computed from ***E(n) = Q(n) − Qref*** where (***Qref***) is the target queue size [15, 29].

## 3.5 Modified Random Early Detection (ModRED)

ModRED algorithm improves the average queue size, on a way limit delay and packet loss rate. The ModRED aims to restrict the TCP transmission window with the flow control window instead of the congestion control window [17]. ModRED treatment the congestion occurring at both the ingress and gateway. ModRED algorithm avoids the dropping of packets at the queue [17] by set the window field to one maximum segment size (MSS) in the ACK packets 1 that are going towards the sender from the receiver, The only exception that it will not modify the receiver window field is when the field has a value of 0. This occurs when a TCP application wants to tell its peer not to send any more data. The field is set to 1 only if the average queue length is between ***Thmin*** and ***Thmax*** [17].

# 4. OUR PROPOSED ALGORITHM

Our proposed algorithmcalled Enhanced Random Early Detection ***(ENRED)***aims to provide better congestion control over the network while preserving the advantage of RED. The algorithm depends on enhancement of the average queue size on a way that limits queue size to minimize the delay and packet loss rate as compared to RED queue, as the ENRED works to make the queue more stable. Average queue size calculation is taking place in the low pass filter in an exponential weighted moving average (EWMA) as shown in equation 1.

It depends on the queue weight parameter ($w_q$) (i.e., the queue weight is determined by the size and duration of bursts in queue size that are allowed at the gateway) considering the time constant of the low pass filter. The ENRED take a new parameter beside the $w_q$ which is called target queue (***qt***) (i.e., the difference between the current queue size and the average of the maximum threshold and minimum threshold). If the target queue does not exceed the critical point which is before buffer overflow,***ENRED*** can calculate the average queue size according to the following algorithm:

***Target = ( MAXth + MINth) / 2;***
***Every*** $q_{avg}$ ***update:***
***for each arrival packet before the bufferoverflow***
***if (*** $q_{avg}$ ***< q(size) < critical(th) )***
***qt =q – target;***
$q_{avg} = qt\ (1- w_q) + q.\ (qt\ - w_q)$ ***;***

In our proposed (ENRED),the average queue sizeis calculated by the equation 3 that is the modification ofthe equation 1. By this equation, the performance is more enhanced than before.

$$q_{avg} = qt\ (1- w_q) + q.\ (qt\ - w_q) \text{ ---------- (3)}$$

This paper shows the comparison between three algorithms: RED algorithm, ModRED algorithm and our proposal (ENRED) algorithm because RED is the main and original algorithm in this issue, and ModRED is a new algorithm which appeared in the past few years.

# 5. PERFORMANCEEVALUATION

## 5.1 Evaluation metrics

The evaluation metrics are: ***(1)Queue size***, which shows the periods of buffer underflow and overflow. ***(2)Delay***, is the required time of two way communication, it ranges within a very few microseconds, and can be measured as per packet transfer times. ***(3)Packet losses***, which refer to the number of dropping packets per unit of time. It may also be defined as the packets that are retransmitted again from the source because the packet is either corrupted or lost. ***(4)Congestion window*** is a flow control imposed by the receiver. The former is based on the sender's assessment of perceived network congestion and the latter related to the amount of available buffer space at receiver for this connection.

## 5.2 Simulation setup

The simulation is often used for understanding and prediction of behavior of protocols and data streams in the network. The results are obtained using network simulator 2 (NS2) [31]. The topology used 50 connections and bottleneck congestion. The bottleneck link bandwidth is 3 Mbps and the transmission time of data from sender to receiver is *100 ms* the gateway set $w_q$=0.002, minimum threshold=15, maximum threshold=45 and maxp=1/50. Table (1) shows the ENRED parameters.

**Table 1. ENRED Parameters**

| | |
|---|---|
| **Bandwidth of bottleneck link** | 3 Mbps |
| **Propagation delay of bottleneck link** | 100 ms |
| **Packet size** | 1024 byte |
| **Buffer size** | 100 packets |

## 5.3 Simulation results

The test show the evaluation of RED, ModRED and ENRED algorithms in the figures (1 to 4) which show the queue size, the delay, and the packet loss rate, and congestionwindow versus simulation time, respectively. It is noted that Figure (1) shows the queue size of each algorithm with 50 TCP connections which show the periods of buffer underflow and overflow and the queue size. It can be seen that the queue size of RED is unstable, due to the simulation, where ModRED and ENRED are stable around the buffer occupancy.

Figure (2) shows the delay of packet in the queue for each algorithm, it is noted that ENRED achieves more predictable packet delay than others. This refers to achieving the stabilized queue size with target queue. Although the queue size of both ModRED and ENRED are approximately the same, but the packet delay in ENRED is less than that in ModRED because the congestion window is a little more in ENRED than that in ModRED.

Figure (3) depicts the packet losses of each algorithm with 50 connections. Among them, the packets loss rate of RED is the highest, but ENRED is achieving low packet loss by 33 % and this refers to the stabilized queue and the packet delay of ENRED is less that the packet delay of ModRED. It means that which having low delay, having low losses.

Figure (4) shows the congestion window of each algorithm which show that ModRED and ENRED have small congestion window increasing its window in congestion avoidance, but RED repeatedly is suffering packet drops, so it can pick the same connection from which to drop packets for a short period of time, causing temporary non-uniform dropping among identical flows.

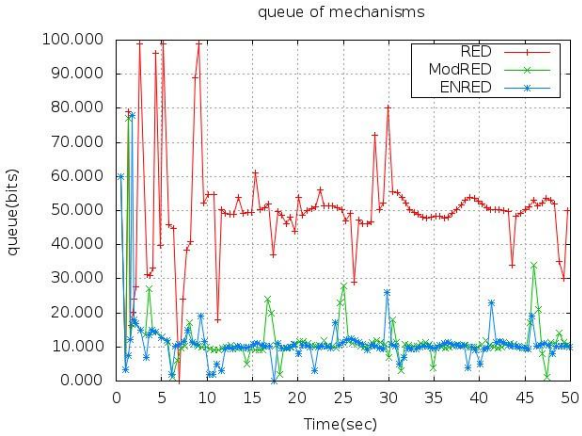


**Fig1: queue size versus time for RED, ModRED, and ENRED.**

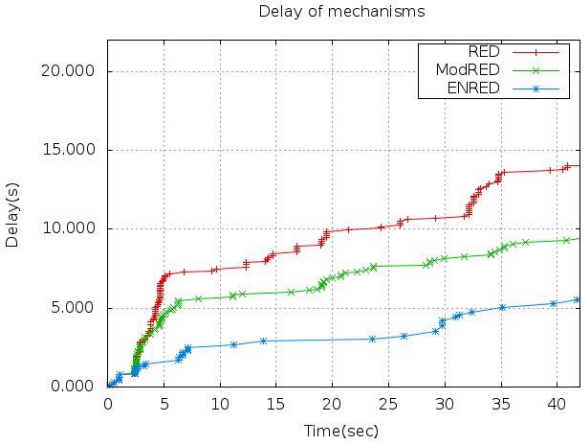


**Fig2: delay versus time for RED, ModRED, and ENRED.**

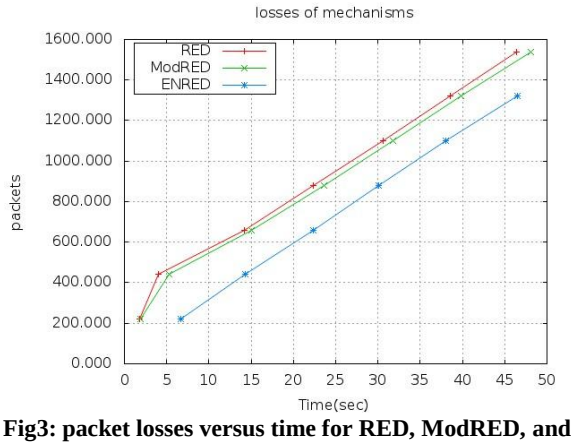


**Fig3: packet losses versus time for RED, ModRED, and ENRED.**

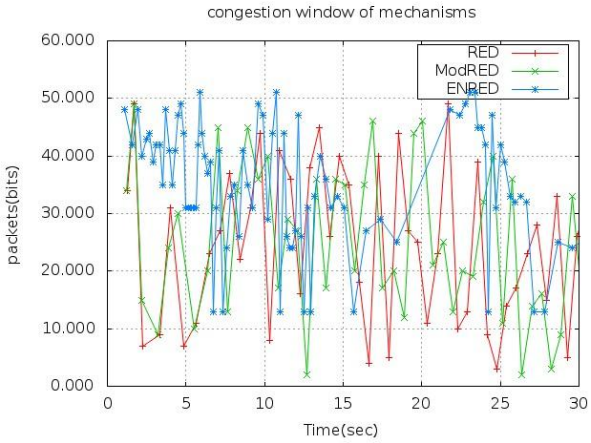


**Fig4: congestion window versus time for RED, ModRED, and ENRED.**

## 6. CONCLUSION

This paper presented an overview about the congestion control mechanism and concentrate on the RED algorithm and its variants and their important roles in congestion avoidance, including our proposed algorithm. The simulation results are related to RED, ModRED, ENRED algorithms. It has proposed an enhancement to existing RED algorithm called ENRED which does not require modification to end system. This scheme helps to reduce the average queue size of the RED queue. ENRED results in small queue size which leads to less delay and low packet loss rate.

## REFERENCES

[1] Ayman EL-SAYED, Nawal EL-FESHAWY and Shimaa HAGAG, "A survey of Mechanisms for TCP Congestion control", The International Journal of Research and Reviews in Computer Science (IJRRCS'11),pp. 676-682, Vol.2,No.3, June 2011.

[2] M. Kalpanal and T. Purusothaman, "Performance Evaluation of Exponential TCP/IP Congestion Control Algorithm", International Journal of Computer Science and Network Security (IJCSNS), VOL.9 No.3, March 2009.

[3] Seifeddin Kadry, Issa Kamar, Ali KalaKech, Mohamed Smaili Robust, "TCP: An Improvement on TCP Protocol", Journal of Theoretical and Applied Information Technology 2005.

[4] Shimaa HAGGAG, Ayman EL-SAYED, "Enhanced TCP Westwood Congestion Avoidance Mechanism (TCP WestwoodNew)", International Journal of Computer Applications, vol. 45, No. 5, pp. 21-29, May 2012.

[5] P.Radhadevi, Ch.Srikanth, "Congestion Avoidance by TCP", International Journal of Computer Trends and Technology(IJCTT) –volume4, Issue6–June2013.

[6] Nga J.H.C., Iu H.H.C., Ling S.H., Lam H.K. "comparative study of stability in different TCP/RED models", Chaos, Solitons and Fractals, The interdisciplinary Journal of Nonlinear Science, and Nonequibrium and complex phenomena, Vol.37, Issue 4, August 2008, pp.977-987.

[7] P.Kuusela P. Lassila, J. Virtamo and P. Key, "Modeling RED with Idealized TCP Sources", 9th IFIP Conference

on Performance Modeling and Evluation of ATM & IP network, 2001.

[8] S.H Low, F. Paganini, and J.C. Doyle, "Internet Congestion Control", IEEE Control Systems Magazine, FEB 2002, PP: 28-43.

[9] L.Yao-Nan, and H. Ho-Cheng, "A New TCP Congestion Control Mechanism over Wireless Ad Hoc Networks by Router-Assistant Approach", International Conference on Distrubuted Computing Systems, Jun 2007, PP: 84-84.

[10] S.Ryu, C.Rump, and C. Qiao, "advances In internet Congestion Control", IEEE Communications Surveys & Tutorials, Third Quarter, Vol.5(1), 2003, PP:28-39.

[11] C.Wanxiang, S. Peixin, and L.Zhenming, "Network-assisted congestion control", Info-tech&Info-net Iternational Conferences, Vol.2, JUN 2001, PP:28-32.

[12] K Fang-Chun, and X. Fu, "Probe-Aided MulTCP: an aggregate congestion control mechanism", ACM SIGCOMM Computer Communication Review, Vol.38 (1), JAN 2008, PP: 17-28.

[13] Andrzej Chydzinski, Agnieszka Brachman, "Performance of AQM Routers in the presence of New TCP Varints", Advanced in Future Internet, International Conference on, PP. 88-93, 2010 Second International Conference on Advances in future Internet, 2010.

[14] Chengyu Zhu, O.W.W. Yang, J. Aweya, M. Ouellette, Montuno, A comparison of active queue management algorithms using the OPNET Modeler, Communications Magazine, IEEE, volume 40, Pages: 158 – 16, June 2002.

[15] Victor Firoiu, Marty Borden , A Study of Active Queue Management for Congestion Control, Nortel Networks, Billerica, MA, INFOCOM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings. IEEE, Volume 3, Pages: 1435 - 1444 vol.3, Mar 2000.

[16] Wu-chang Feng,"IMPROVING INTERNET CONGESTION CONTROL AND QUEUE MANAGEMENT ALGORITHMS", A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, (Computer Science and Engineering) in The University of Michigan, 1999.

[17] G.A. Ramachandra, Reshma Banu and G.F.Ali Ahammed, Analyzing Marking Mod RED Active Queue Management Scheme on TCP applications, 2012 International Conference on Information and Network Technology (ICINT 2012) IPCSIT vol. 37 (2012) © (2012) IACSIT Press, Singapore.

[18] W.Boulevard, and A.way, "Transmission Control Protocol", RFC 793, September 1981.

[19] V.Jacobson and M.J.Karels, "Congesion avoidance and control", In ACM computer Communication Review; Proceeding of the Sigcomm'88 Symposium, volume 18, pages 314-329, Standford, CA,USA, Augut 1988.

[20] Hanaa A.Torkey, Gamal M. Attiya and I. Z. Morsi,"Performance Evaluation of End-to-End Congestion Control Protocols", Menofiya Journal of Electronic Engineering Research (MJEER), Vol.18, No.2, PP. 99-118,July 2008.

[21] Jianyong Chen, Cunying Hu, and Zhen Ji , Self-Tuning Random Early Detection Algorithm to Improve Performance of Network Transmission, Hindawi Publishing Corporation Mathematical Problems in Engineering Volume, Article ID 872347, 17 pages doi:10.1155/2011/87234, 2011.

[22] G.F.Ali Ahmed, Reshma Banu, Analyzing the performance of Active Queue Management Algorithms, International journal of Computer Networks & Communications (IJCNC), Vol.2, No.2, March 2010.

[23] Chandra Himanshu, Agarwal Ajay, Velmurugan T, "Analysis of Active Queue Management Algorithms & Their Implementation for TCP/IP Networks Using OPNET Simulation Tool", International Journal of Computer Applications (IJCA), Vol. 6, Issue 11, Sept. 2010, pp. 12-15.

[24] S.Dijkstra, Modeling Active Queue Management algorithms using stochastic Petri Nets, Faculty of Electrical Engineering, Mathematical and Computer Science, University of Twente, December 14, 2004.

[25] S.Floyed, V.Jacobson, Random Early Detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking August 1993.

[26] Michael Welzl, Leopold Franzens Network Congestion Control Managing Internet Traffic, University of Innsbruck.

[27] R. J. La, P. Ranjan, and E. H.Abed, Analysis of Adaptive Random Early Detection (ARED), Networking, IEEE/ACM Transaction, Volume 12, Pages: 1079 – 1092, 2004.

[28] Sunitha Burri, BLUE: Active Queue Management CS756 Project Report, May 5, 2004.

[29] Teresa Álvarez, Virginia Álvarez, Lourdes Nicolás, UNDERSTANDING CONGESTION CONTROL ALGORITHMS IN TsCP USING OPNET, Spain, 2010.

[30] 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.

[31] NS-2 network simulator (ver. 2). LBL, URL:http://www.mash.cs.berkeley.edu/ns.