EPH - International Journal of Science And Engineering
ISSN (Online): 2454-2016
Volume 03 Issue 04 December 2017

DOI: https://doi.org/10.53555/eijse.v3i4.111

IMPLEMENTATION OF THE RED MECHANISM FOR CONGESTION AVOIDANCE USING C++ LANGUAGE

Haytham Ahmed[1*]
Information Technology, Faculty of computer &information Technology, University of Jeddah, KSA.

Corresponding author:-
E-mail:- HABDELMEGED@uj.edu.sa , borena100@hotmail.com, borena1000@gmail.com

Abstract:-
*Congestion lowers the network utilization, throughput which causes very large queuing delays, data loss and degrades quality of service. In this paper the technique **R**andom **E**arly **D**etection (RED) has been studied in more details and other queuing techniques have been derived in general in order to solve congestion problem in the computer networks.*
*The simulation program has been built using C++ language to simulate **R**andom **E**arly **D**etection (RED) algorithm which gave a detailed result about the algorithm.*

Keywords: - ***R**andom **E**arly **D**etection,*

1. INTRODUCTION

Usually, allocating the network hosts to a small logical network according to the operation unit of the organization. In the way, most of the traffic of these hosts can be restricted in a relatively small range, thus reducing the influence on other hosts and reducing the load of the network trunk. Base on this division, the data traffic mode in the traditional network follows the 80/20 rules [1].

The 80/20 traffic model means that 80% of the user's data traffic is at the local network segment, while only 20% enters other network segments via routers or bridges refer to figure (1-1).

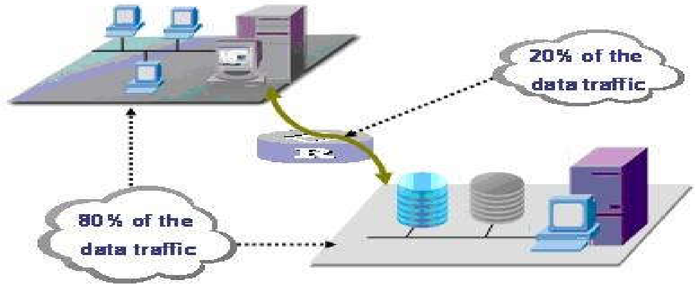
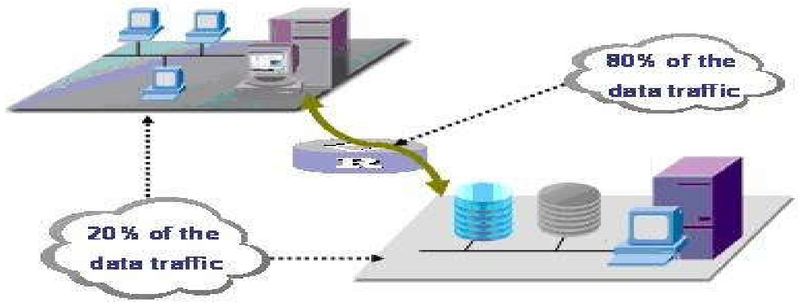


Figure 1-1 traditional 80/20 traffic models

Source: Huawei technologies Co./ VRP Operation Manual

2. For the network adopting the 80/20 rule, most of the user's network resources are in the same network segment. These resources include the network server, printer, shared directory and files. If more than 20% of the traffic strides over network segments, performance problem may arise [1].
3. With the applications of INTERNET/INTRANET and the emergence of the distributed servers, changes have been made to the traditional 80/20 traffic model. In the network, most of the data flows through the backbone, and there is not a big data stream inside the logical subnet. And the user needs to frequently access the resources outside the local subnet. The new 20/80 rule just adapts to the new traffic mode [1], refer to figure (1-2).



4. In addition, managing application performance can be quite a challenge. Productivity drops and frustration climbs when performance turns inconsistent, unpredictable, and slow.
5. For many companies, business application performance declined gradually from sufficient to unworkable. In other organizations, a single event, such as deploying a new application or relocating servers, seems to precipitate the decline. Poor network and application performance can be addressed.
6. With today's enterprise network infrastructure becoming more complex and dynamic, a successful network management requires more than knowing if a device is available. The NMS must ensure that the network and applications are up-and-running smoothly and efficiently. It must also proactively identify, diagnose and predict potential problems before they have any impact on the network. Recognizing the challenges of enterprise-grade networks, an effective testing and evaluation of network management system (NMS) becomes essential [2].

3. In high-speed networks with connections with large delay-bandwidth products, gateways are likely to be designed with correspondingly large maximum queues to accommodate transient congestion. In the current Internet, the TCP transport protocol detects congestion only after a packet has been dropped at the gateway [10].
This Chapter presents Random Early Detection (RED) gateways for congestion avoidance and buffer management in packet switched networks.

The gateway detects incipient congestion by computing the average queue size. The gateway could notify connections of congestion either by dropping packets arriving at the gateway or by setting a bit in packet headers. When the average queue size exceeds a preset threshold, the gateway drops each arriving packet with a certain probability, where the exact probability is a function of the average queue size [9].

RED gateways keep the average queue size low while allowing occasional bursts of packets in the queue. During congestion, the probability that the gateway notifies a particular connection to reduce its window is roughly proportional to that connection's share of the bandwidth through the gateway.

RED gateways are designed to accompany a transport-layer congestion control protocol such as TCP.

The RED gateway has no bias against bursty traffic and avoids the global synchronization of many connections decreasing their window at the same time. Simulation is used to illustrate the performance of RED gateways.[10]

4.1. Red algorithm

The RED gateway calculates the average queue size, using a low-pass filter with an exponential weighted moving average. The average queue size is compared to two thresholds, a *minimum* threshold and a *maximum* threshold. When the average queue size is less than the minimum threshold, no packets are dropped. When the average queue size is greater than the maximum threshold, every arriving packet is dropped. If marked packets are in fact dropped, this ensures that the average queue size does not significantly exceed the maximum threshold [10].

When the average queue size is between the minimum and the maximum threshold, each arriving packet is marked with probability p_a, where p_a is a function of the average queue size *avg*. Each time that a packet is marked, the probability that a packet is marked from a particular connection is roughly proportional to that connection's share of the bandwidth at the gateway. The general RED gateway algorithm is given in Figure (4-1) [10].

The RED gateway has two separate algorithms. The algorithm for computing the average queue size determines the degree of burstiness that will be allowed in the gateway queue. The algorithm for calculating the packet-marking probability determines how frequently the gateway marks packets, given the current level of congestion [10].
The goal is for the RED gateway to mark packets at fairly evenly-spaced intervals, in order to avoid biases and to avoid global synchronization, and to mark packets sufficiently frequently to control the average queue size.

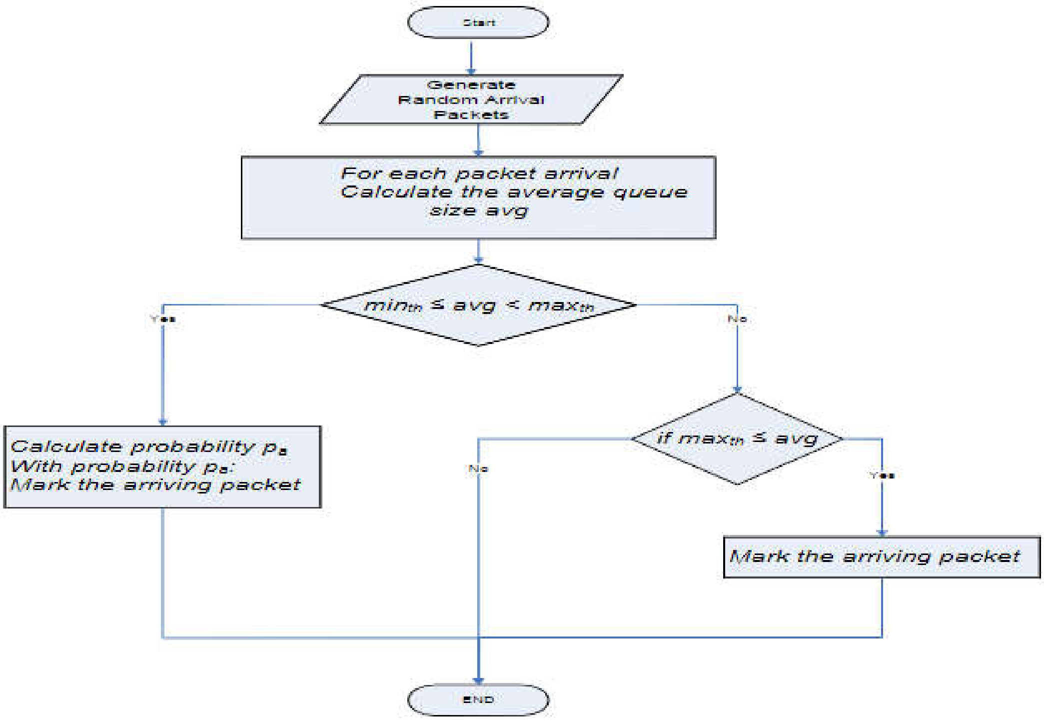


Figure (4-1): Flow chart for general algorithm for RED gateways.

Source: Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1

4.2. Calculating the average queue length

The RED gateway uses a low-pass filter (exponential weighted moving average) to calculate the average queue size. Thus, the short-term increases in the queue size that result from bursty traffic or from transient congestion do not result in a significant increase in the average queue size [10].

$$\text{Avg} = (1-w_q)\,\text{avg} + w_q\,q \;\rightarrow \text{The low-pass filter}$$

Where queue weight w_q is the time constant of the low-pass filter, q is the queue size and avg is the average of the current queue size.

The optimal values for min_{th} and max_{th} depend on the preferred average queue size. If the typical traffic is fairly bursty, then min_{th} must be correspondingly large to allow the link utilization to be maintained at an acceptably high level. For the typical traffic in this simulation, for connections with reasonably large delay bandwidth products, a minimum threshold of one packet would result in unacceptably low link utilization [11].

The optimal value for max_{th} depends in part on the maximum average delay that can be allowed by the gateway.

RED gateway functions most effectively when $max_{th} - min_{th}$ is larger than the typical increase in the calculated average queue size in one roundtrip time. A useful rule-of-thumb is to set max_{th} to at least twice min_{th}.

The initial packet-marking probability p_b is calculated as a linear function of the average queue size.

$$p_b = max_p\,(avg - min_{th})/(max_{th} - min_{th}).$$

The parameter max_p gives the maximum value for the packet-marking probability p_b, achieved when the average queue size reaches the maximum threshold.

4.3. Simulation programs

The basic objective of the simulator is to find evaluate the performance of the random early detection technique in avoiding congestion and manage the queue size for packet switched network. In addition, to illustrate the configuration of QoS on Cisco routers, debug the configuration and show results.

4.3.1. A C++ simulator

This is the main simulator to get result that can be used for efficiency evaluation, figure (4-2) shows Simulation program flow chart.

Based on RED algorithm a program has been written using C++ languages (refer to a simulator code in appendix A), the C++ simulator program divided to many functions, each function perform specific task, then those functions are going to be called in the main function to do the RED task.

The clock used in this simulation is a simulation clock, which adjustable whenever there is an action (arrival/depature event) occurs.

The RED parameter [16] values which are used in this simulation are the values that used in the original RED algorithm [10], where $w_q = 0.002$, $min_{th} = 5$ packets, $max_{th} = 15$ packets, $max_p = 1/50 = 0.02$ and pkrate = 12000 (~1K packets on 100Mbps link).

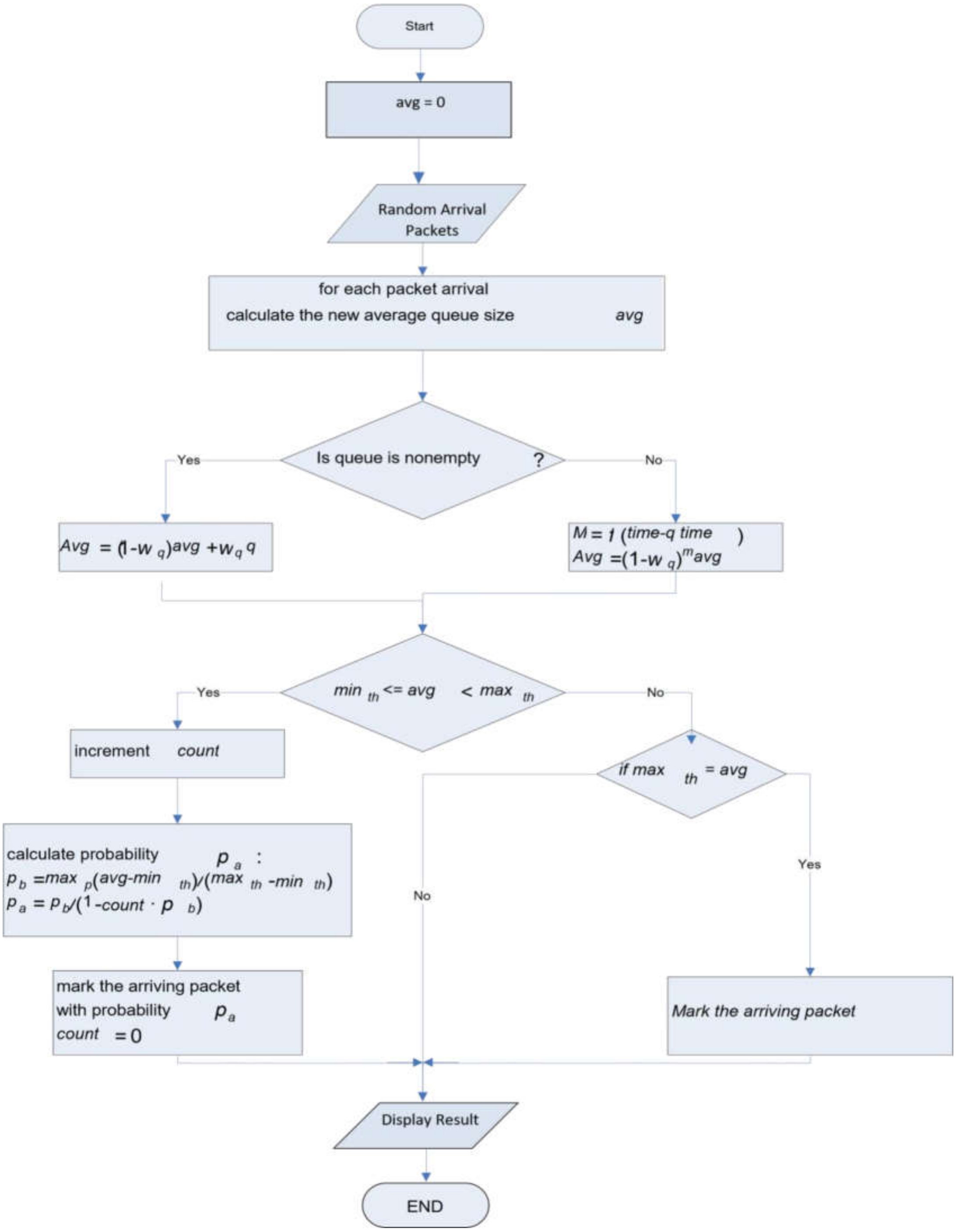


Figure (4-2): RED Simulation program flow chart

RED program function's as the following

1. Initialization function is responsible for setting the initial values such as simulation clock, queue size, sent & received packages and total package lost which are going to be used by the program.
2. Event list function is responsible for generating events and creates the starting arrival time of the simulation.
3. Scheduler function is responsible for records the event type, either send or receive.
4. Clock function is responsible for the simulation time.
5. Swap function is responsible for moving the packets into the front of the queue.
6. Calculate average function used to calculate queue size average.
7. Arrival function responsible for scheduling the next packet, check the service counter, increase the queue size and register arrival time for each packet place in the queue.
8. Swap function is responsible for moving the Packets in Q to the front of the queue.
9. Departure function is responsible for increment the total number of packets departed, reduce the Q size as the customer moved out of the queue, calculate the delay for every customer and calculate the total delay of the system.
10. Result function is responsible for print the results of the system into standard output.
11. Main function is used to call all sub-functions which are cooperative together to give the simulator objectives.

4.3.2. Real router simulator

This simulation is a real router simulator i.e. CISCO internetwork operating system (IOS) software is used to simulate the real router, Cisco 7200 series is the selected router. Figure (4-3) shows the simulator topology (refer to the simulator configuration in appendix B).

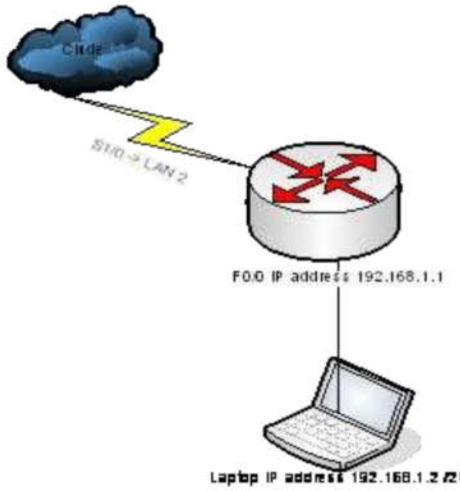


Figure (4-3) simulator topology

Building the simulator

DYNAMIPS is a Cisco router emulator written by Christophe Fillot. It emulates 2691, 3620, 3640, 3660, 3725, 3745, and 7200 hardware platforms, and runs standard IOS images.

This kind of emulator would be useful to be used as a training platform, with software used in real world. It would allow people to become more familiar with Cisco devices, Test and experiment features of Cisco IOS and check quickly configurations to be deployed later on real routers. Refer to appendix (B) for configuration file and configuration commands.

4.4. Results of RED gateways

The results of the RED congestion avoidance simulator program are used to draw the following graphs using MS-Excel, refer to appendix (c).

By using RED congestion avoidance technique, normalized throughput of the network is stable. If Maxp parameter equal 0.1, throughput will become more stable than throughput using Maxp equal 0.02, if RED algorithm is not used then throughput become very week, refer to figure (4-4 (a,b)).

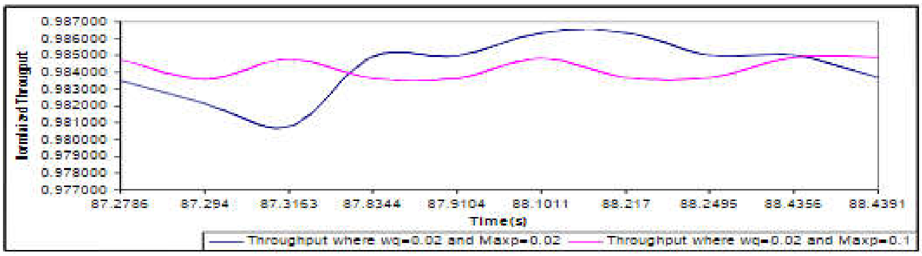


Figure (4-4, a) RED Normalized throughput Vs Time Source: Simulation output results

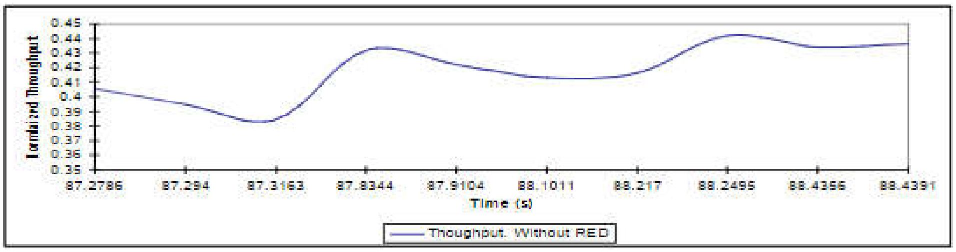


Figure (4-4,b) WITHOUT RED Normalized throughput Vs Time (b) Source: Simulation output results

Without using RED the packets loss ratio is high, but by using RED the packets loss ratio reduced, refer figure (4-5, a).

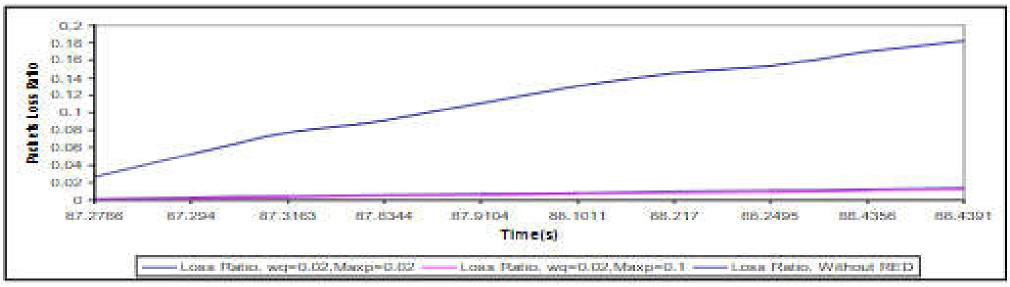


Figure (4-5,a) Packets loss ratio (With/Without RED) Vs time Source: Simulation output results

By changing RED parameter (Maxp) form 0.02 to 0.1, packets loss ratio deceases that means the C++ simulator give results as the default in the ns-2 simulator which is now set max_p to 0.1 refer figure (4-2, b).

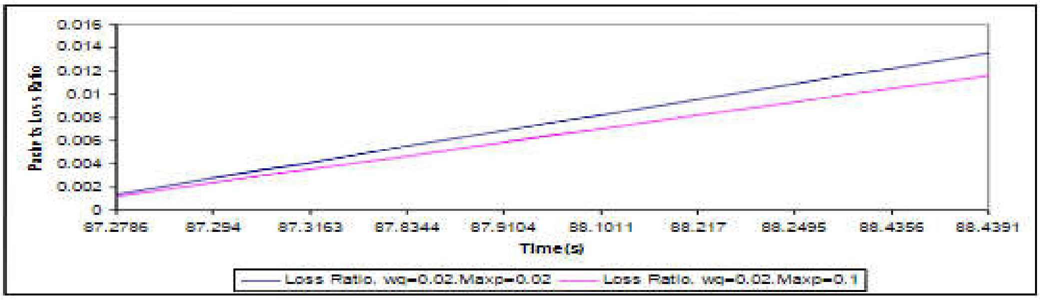


Figure (4-5,b) RED Packets loss ratio Vs Time

Source: Simulation output results

Usually the average delay increase with incensement of the dropped packets during congestion, by implementing RED average delay is reduced refer to figure (4-6,a,b and c)

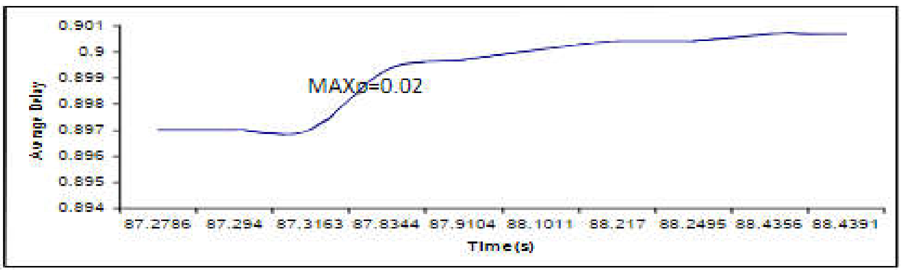


Figure (4-6,a) Average Delay Vs Time (a) Source: Simulation output results

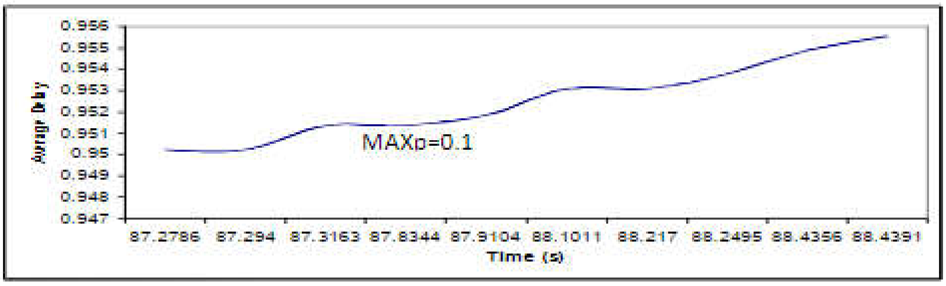


Figure (4-6,b) Average Delay Vs Time (b) Source: Simulation output results

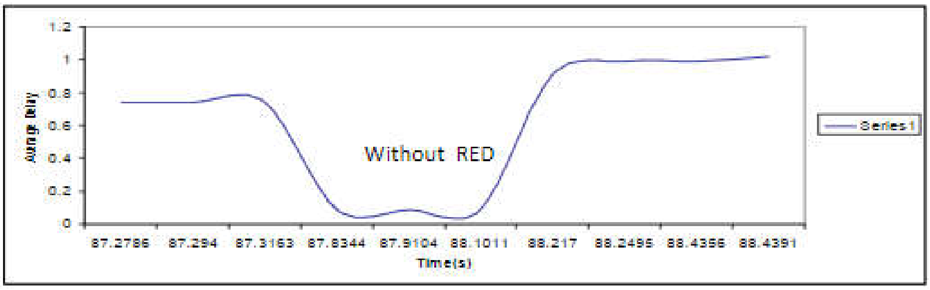


Figure (4-6,c) Average Delay Vs Time(c) Source: Simulation output results

5. Conclusion

In general, congestion is the major problem in network environment, to control network traffic it is very important to apply a comprehensive mechanism to avoid congestion before it occur and if it happens there should be ways to manage this congestion.

Random Early Detection gateways are an effective mechanism for congestion avoidance at the gateway, in cooperation with network transport protocols. If RED gateways drop packets when the average queue size exceeds the maximum threshold, then RED gateways control the calculated average queue size.

For RED gateways, the rate at which the gateway marks packets depends on the level of congestion, avoiding the global synchronization that results from many connections decreasing their windows at the same time.

RED gateway is a relatively simple gateway algorithm that could be implemented in current networks or in high-speed networks of the future. RED gateway allows conscious design decisions to be made about the average queue size and the maximum queue size allowed at the gateway

References

[1].R&D Huawei technologies Co./ Versatile Routing Protocol, Operation Manual QoS Volume 5.10/ October 2006.p. 11.

[2].William Stallings/ Data and computer communication, 6th Edition/ prentice hall press 2000/pp 65-69.

[3].Raj Jain/ "Myths about congestion management in high speed networks Internetworking/ research and experience, Vol 3, 1992, pp. 101-113.

[4].Answers Corporation/ http://www.answers.com/topic/bandwidth?cat=biz-fin / 2000/March 2007.

[5].Answers Corporation/ http://www.answers.com/topic/network-delay / 2000/March 2007.

[6].Answers Corporation/ http://www.answers.com/topic/traffic-policing?cat=technology / 2000/March 2007.

[7].Bala, K., Cidon, I., and Shrubby, K./"Congestion Control for High Speed Packet Switched Networks"/ INFOCOM '90, pp. 520-526, 1990.

[8].Floyd, S./ Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One way Traffic/ Computer Communication Review, V.21 N.5/ October 1991, pp. 30-47.

[9]. Hashem, E./ "Analysis of random drop for gateway congestion control"/ Report LCS TR-465, Laboratory for Computer Science/ MIT, Cambridge, MA, 1989/ p.103.

[10]. Mitra, D. and Seery, J./ Dynamic Adaptive Windows for High Speed Data Networks: Theory and Simulations/ Proceedings of SIGCOMM '90/ September 1990, p.30-40.

[11]. Romanow, A./ "Some Performance Results for TCP over ATMwith Congestion"/ to appear in Second IEEEWorkshop on Architecture and Implementation of High Performance Communications Subsystems (HPCS 93)/ Williamsburg, VA, Sept. 1-3, 1993.

[12]. Floyd, S./ "Issues in Flexible Resource Management for Datagram Networks"/ Proceedings of the 3rd Workshop on Very High Speed Networks/ March, 1992.

[13]. ICIR (The ICSI Center for Internet Research), http://www.icir.org/floyd/REDparameters.txt / 2007/ Oct 2007.

Mr.HAYTHAM MERGANI AHMED , PHD **in information Technology MSc(2009) in information Technology** at Faculty of Computer Science and Information Technology – AL Neelain University - Khartoum – Sudan, Assistant Professor at University of Jeddah- king audia Arabia - Elimam Elhadi collage- Sudan - mobile :+249912981116 - +966507864981.