MRED: An Algorithm to Insure High QoS in IP Networks

El-Bahlul Fgee[1], Adel Smeda [2], and Khadija AbouElgaseem[3]
[2]Department of Computer Science, High Institute of Vocational Studies, Yafren, Libya
[1]Department of Data Analysis, Faculty of Accounting, UniversityAl-Jabal Al-Gharbi, Gharian, Libya
[3]Department of Computer Science School of Basic Sciences, Libyan Academy, Trpoli, Libya
Email: fgeeee@dal.ca; Adel.smeda@jgu.edu.ly; khadija.abou@gmail.com

Abstract—Today's computer networks support only best-effort service, yet future packet networks will have to support real-time communication services that allow clients to transport information with performance guarantees expressed in terms of delay, delay jitter throughput, and loss rate. Random Early Detection (RED) algorithm is one of the scheduling mechanisms used in IP network. RED is an active queue management scheme that randomly drops packets whenever congestion occurs that results in dropping traffic flow packets that are sensitive to loss. This results in QoS degradation. However, Quality of Service (QoS) is a major factor for a successful business in modern and future network services. In this article, the RED algorithm is modified in order to minimize the packet loss of sensitive traffic flows. Evaluation of the proposed enhancement of the RED algorithm is tested using the Network Simulator (NS-2). Then these results are compared with the current RED algorithm. The proposed model shows ten improvements in QoS of sensitive traffic flows.

Index Terms—QoS, RED, MRED, packet loss

I. Introduction

Modern communication networks require integration of a variety of data flows into an infrastructure that may not be specifically suited to handle the requirements and characteristics of the traffic. Under heavy loads, decisions must be taken concerning which packets will be discarded in order to maintain stability in the network. In the RED (Random Early Detection) mechanism, any packet could be, RED is the active queue management scheme which randomly drops the packets whenever congestion occurs, however RED works in dropping packets are not equal between different types of data, i.e. may pass through a same link and as a result losses in the these packets of data could be too large. For packets, that use TCP protocol this problem can be overcome by retransmitting the lost packets, while the process becomes more complicated for real-time applications that use UDP protocol because of the sensitivity of this type of packets is lost and delays as well. As pointed out RED allows unfair bandwidth sharing when a mixture of the three traffic types shares the same link. This unfairness is caused by the fact that at any given time RED imposes the same loss rate on all flows, regardless of their bandwidths. In order to deal with the existing problem, we tried to enhance RED algorithm. The currently policies used now in RED queue management do not solve the drop problem so a new policy is added to the enhanced RED to solve problem of high drop rate [1].

The objective of this article is to introduce an enhancement to the RED algorithm, which will have two virtual queues for each traffic type, TCP or UDP. The enhanced RED algorithm will drop data packets only during congestion and this will result in high QoS. This algorithm will be evaluated using network simulator -2 and compared with two types of policies.

II. Active Queue Management

Active Queue Management (AQM) refers to a family of packet dropping mechanisms for router queues that has been proposed to support end-to-end congestion control mechanisms in the Internet. These mechanisms were design for TCP congestion which relies on the network to signal congestion by dropping packets. If the routers use traditional FIFO queuing, the buffer will full up before the TCP senders reduce their transmission rate. The main idea behind the AQM is to get TCP senders reduce their transmission rate early [2]. Before the queues have become over full, the router will signal TCP sender by dropping or marking packet at randomly. [3].

A. Packet Drop Schemes (Drop Tail)

First-in-first-out or drop-tail-when-full was the original queue management scheme used in Internet routers. With this scheme packets are en-queued at the tail of a queue as they arrive and dequeued from the head of queue when there is capacity on the link. Drop Tail is the policy of dropping the arriving packet when the queue is full. Other alternatives include dropping the packet at the head of the queue. The DT queue management mechanism drops the packet that arrives when the buffer is full. However, this method has two drawbacks: first, one, this mechanism allows a few connections with prior request to dominate the queue space allowing other flows to starve making the network slower. Second, one, DT allows queues to be full for a long period. During that period, incoming packets are dropped in bursts. This causes a severe reduction in throughput of the TCP flow [4].

B. Explicit Congestion Notification

Explicit method of signaling congestions was proposed by Ramakrishnan and Jain [5], i.e. Explicit Congestion Notification. This technique uses the congestion-

Manuscript received January 21 2017; revised April 21, 2017.
doi:10.12720/jcm.12.4.200-206

indication bit in IP header to notify the sender when the congestion occurs. This bit provides feedback about congestion in the network. When a packet arrives at the gateway, the gateway calculates the average queue length for the last (busy + idle) period plus the current busy period [6]. When the average queue length exceeds one, then the gateway sets the congestion-indication bit in the packet header of arriving packets.

The explicit congestion notifications used by Congestion avoidance techniques. These schemes monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks. Congestion avoidance is achieved through packet dropping. Among the more commonly used congestion avoidance mechanisms is Random Early Detection (RED), which is optimum for high-speed transit networks.

C. Random Early Detection (RED)

Floyd and jacobsan at 1993 [7] have introduced an Active Queue Management (AQM) mechanism. This mechanism is called Random Early Detection (RED) which is further description in RFC 2309 [8]. RED was proposed to avoid the problem in the drop tail. The basic idea behind RED algorithm is importing can be achieved by aiming to prevent the observed of global synchronization between TCP sessions. Global synchronization is happened when congestion accrue and more one connection start slow start phase. RED schema tries to break global synchronization by keeping track of the average queue and using it as induction rather than actual queue which is used in drop tail schema.

RED has many advantages when we the implantation at router or getaway, one of them is avoiding global synchronization by sent to one sender at time for low rate rather than sent to all senders. Usually, UDP based application can be considered as smooth traffic [9]. At the end host TCP congestion control mechanism is implemented. TCP should respond to packet droppings after a RTT. The upper layer application take care of congestion and further retransmission whereas UDP hosts neglect packet loss and keep pumping data into network. It does not mean that RED algorithm has no impact on UDP application. It has impact on all links of internet traffic, including both TCP and UDP connections.

D. RED Algorithm

Fig. 1 shows how the RED method works and Fig. 2 presents the algorithm [10].

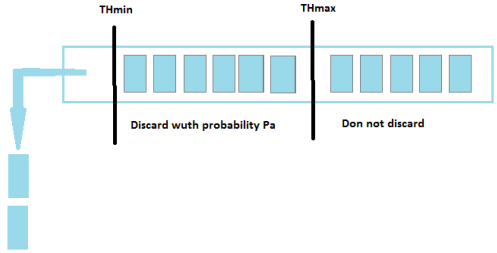

Fig. 1. Random early detection

Calculate the average queue size avg
1 If avg < THmin
Queue packet
2 else if THmin <= avg < Thmax
Calculate probability Pa
3 With probability Pa
Discard packet
4 else with probability 1- Pa
Queue packet
5 else if avg >= THmax

Fig. 2. Pseudo code of TP policy

Although RED scheme has many advantages, researchers found several problems with RED scheme when it is subjected to a mix of traffic. In particular, it may result in lower throughput, larger delay and jitter, and unfairness for some flows at the expense of other flows.

The performance of RED is very sensitive to the maximum drop probability, which determines the aggressiveness of RED towards incoming packets when it is in yellow state. The motivation of varying maximum probability according to the change of average queue size is to lower queuing delay and avoid buffer overflow. Since the build-up of average queue size indicates the imminence of persistent congestion, the RED gateway should be more aggressive when the average queue size increases [11].

III. Enhancement of RED Algorithem

The proposed algorithm takes into consideration the specific characteristics of the real-time traffic and provides a solution that keeps all the advantages of the existing active queue management QoS schemes and reduces as much as possible dropped packets due to buffer overflow. The proposed queuing scheme is able to provide acceptable QoS guarantees by adjusting the queue parameters according to the traffic in environments in which the existing algorithms do not work satisfactorily. In this proposal, two steps are performed, the first step is to divide physical queue into two virtual queues and second step is to add new policy.

As shown in Fig. 3, the virtual queue used as priority, by other mean the first queue used to receive packet with high priority whereas the second queue used to receive packet with low priority. By this step, different traffic will be isolated in two queues. This isolation will make it easier for the core router to choose packet that must dropped when congestion happens.

The policy is implemented by enhancing performance on each packet arriving at the network.

As shown in Fig. 3, the policy is used to compare any two types of packets according to a set of already known rules. In this proposal the policy access to IP header for all packets that arrive from the source node. If this packet uses UDP, it will be given an appropriate code point with high priority where if packet uses TCP, it will be given a different code point with low priority. DSCP or Differentiated Service Code Point shown in Fig. 4 has 8

bits, 6 bits used to mark packet and 2 bits currently unused.

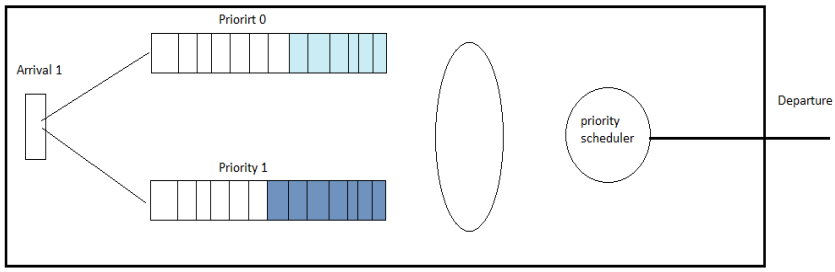

Fig. 3. Using two virtual queues

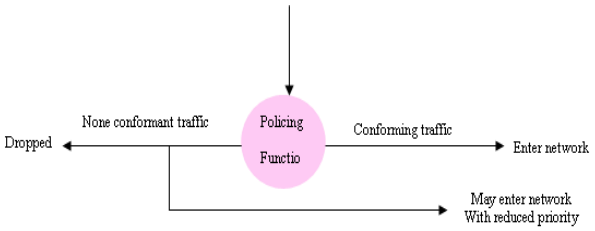

Fig. 4. Policy perform

A. *Implementation of the Enhancement*

This proposal uses code point to mark a packet in green for high priority and in red for low priority. Marked packets can be done at the edge router (after classification), that is used to receive packets and then packets are sent to a appropriate queue. Implementation of the proposed enhancement is divided into four phases:

1) Phase one: Isolated traffic

In this phase queue is divided into virtual queues, one for sensitive traffic and the other one for the traffic with loss tolerance tolerates loss. Fig. 5 shows the flow chart for this phase.

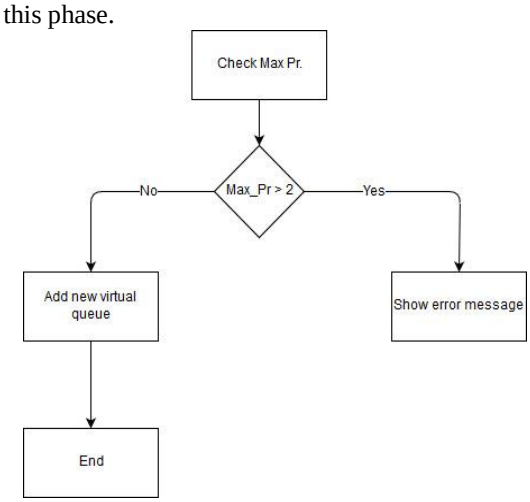

Fig. 5. Flowchart of add virtual queue

2) Phase two: Add new policy

Mark packet with Code Point will be done at edge router by different methods. In this proposal, new method will be added to mark packet based on type of protocol TCP or UDP. NS2 supports many type of traffic such as CBR and FTP which can present in network as real traffic and by the use of popular protocol TCP or UDP to explain how the goal of new proposal achieved.

In diffserv algorithm available two files called "dsPolicy.cc" and "dsPolicy.h", these files contain six types of policies all used to mark packets with different methods. Pseudo code in Fig. 6 and flow chart in Fig. 7 explain and show how to mark a packet with TP policy.

```
Add  new plicy
Define downgrade1 as integer
Define initialCodePt as integer
Define ptype_ as integer

Read ptype_ ;

If ptype_= 0 then

return(policer->downgrade1);

else
return(policer->initialCodePt);
end if
end
```

Fig. 6. Pseudo code of TP policy

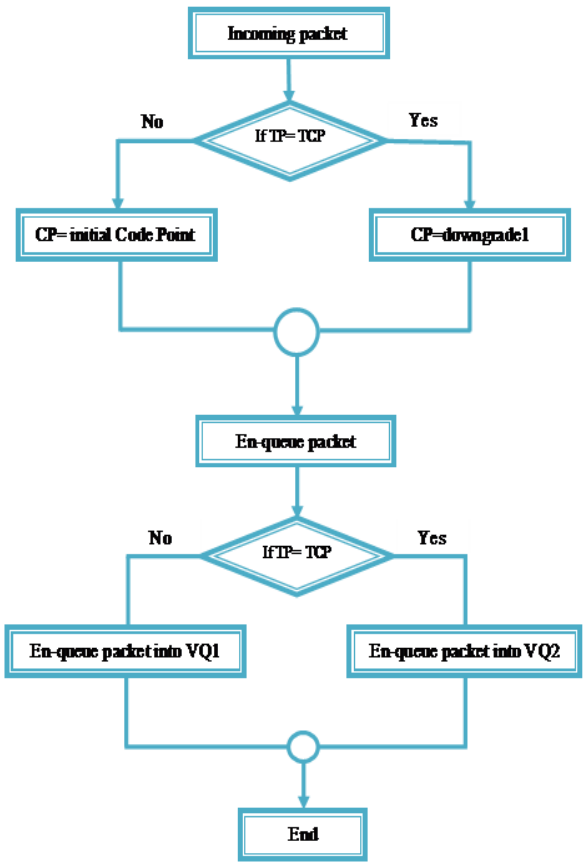

Fig. 7. Flow chart of mark packet with code point

3) Phase three: Congestion avoidance

In MulitRED, congestion happens clearly at core router to avoid this problem, use RED as queue management in two virtual queues. Usually RED starts dropping packet randomly before congestion happens and queue becomes full, while in the MulitRED proposal drops packet selectively using code point. To avoid congestion, core router which placed between two edge routers , start dropping packets with lower priority from virtual queue 1 before packet with high priority, it means UDP an ACK have special treatment packets. Fig. 8 shows a flow chart for service or drop packet at core router.

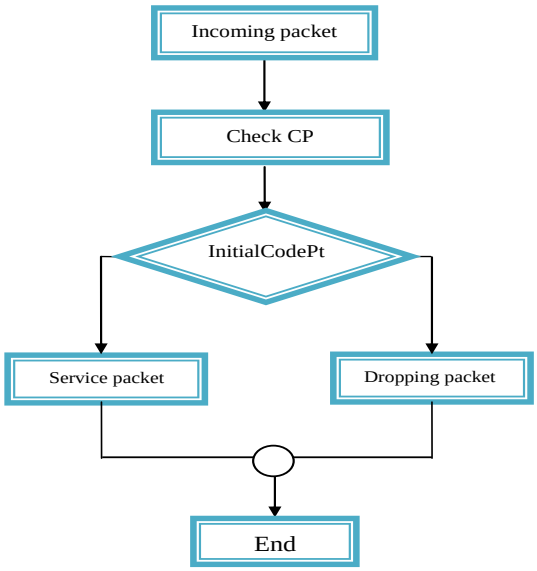

Fig. 8. Flow chart of drop packet

IV. Network Experiments and Simulation Results

Enhanced RED is implemented in Network Simulator version 2, NS-2 [12]. It is used to study the behavior of flow and congestion control schemes in packet switched data networks. Fig. 9 shows the basic architecture of Network Simulator 2.

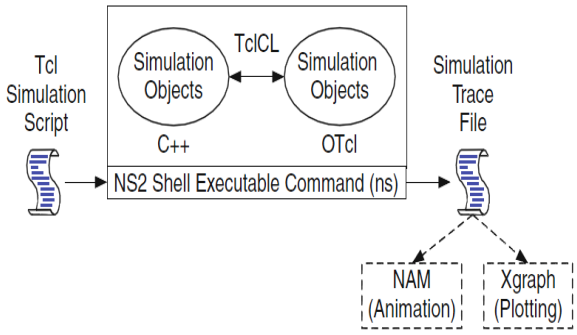

Fig. 9. Basic architecture of network simulator 2

NS-2 is based on two languages, C++ and Object-oriented Tool Command Language (OTcl) interpreter. While C++ defines the internal mechanism of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The OTcl script is processed with the TCL interpreter and NS2 simulator C++ library to provide valid results of the simulation on which an individual works. Fig. 10 shows an example of OTcl script that creates a simple network configuration [13].

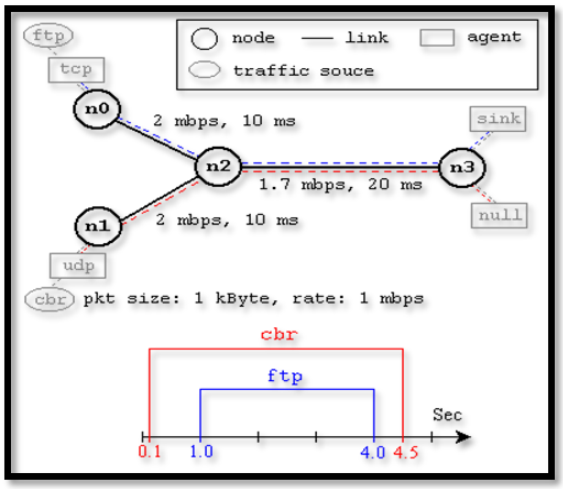

Fig. 10. An example of OTcl script [13].

A. Network Topology Used for the Experiments

The traffic used in the simulation scenarios is with constant bit rate (CBR) and file transfer protocol (FTP). And our transports are TCP, UDP. Drop rate, packet loss rate and through are measured which are all Quality of services. A typical packet size in CBR is 1000 bytes while FTP packet size is 40 bytes. The channel capacity is 10 Mb.

TABLE I: CONFIGURATION OF TWO VIRTUAL QUEUES

Configuration the MRED	Minth	maxth	maxp
Virtual queue1	30	45	0.5
Virtual queue2	10	30	0.1

In order to analyze the drop packet in this proposal by used policy TP, three basic scenarios are simulated:

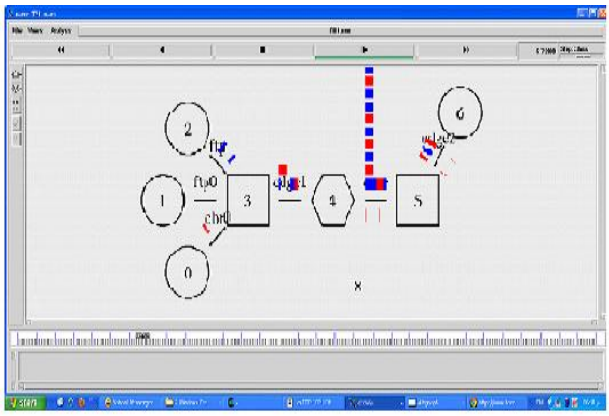

Fig. 11. Results of CBR packets in normal case.

1) Normal case

In first scenario, only three nodes used to generated traffic, one sent CBR packet and two other sent FTP packet. One node used as destination to receive all type of traffic. The simulation showed in Fig. 11 runs during 30.399672s. The results summarized in Table II.

TABLE II: DROPPED CBR PACKET IN NORMAL CASE THREE TYPE POLICIES

Policy type	Packet gen	Packet size	Sent packet	Forwarded packet	Dropped packet	Packet lost	Packet loss rate	Packet loss rate %	Average delay
Token bucket	7691	1000	7691	22984	30	47	0.003900663	0.39007%	0.06118549902
TP	7600	1000	7600	22760	0	22	0	0%	0.09016896912
TSW2CM	7600	1000	7600	22708	30	48	0.003947368	0.39474%	0.06091941102

Fig. 12 shows the impact of the proposal on drop a UDP packet and an ACK packet. In normal case with bandwidth limitation of 10 mbps, no packet drops in both CBR traffic and ACK packet.

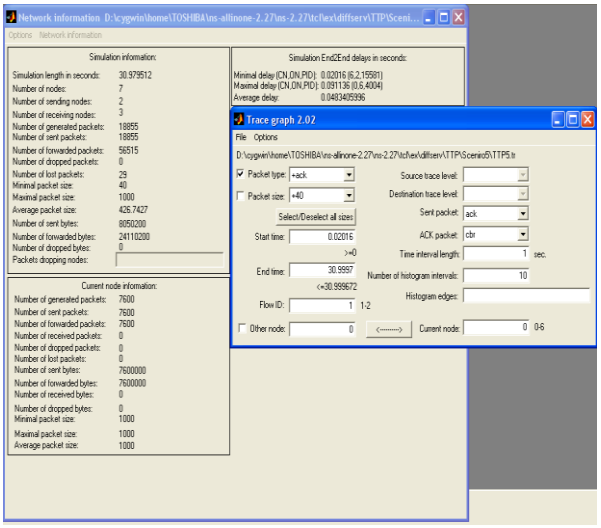

Fig. 12. Results of ACK Packets in normal case with TP

2) Second scenario

Three sources nodes are selected to sent CBR traffic and all other nodes used to generated FTP packet. The process time of all nodes as following: three nodes PC1, PC2, PC3 start to sent CBR traffic respectively 0.6, 0.15, and 0.20 second. Other node, which used to send FTP traffic, start at the same time at 0.0 second. The simulation scenario shown in Fig. 13.

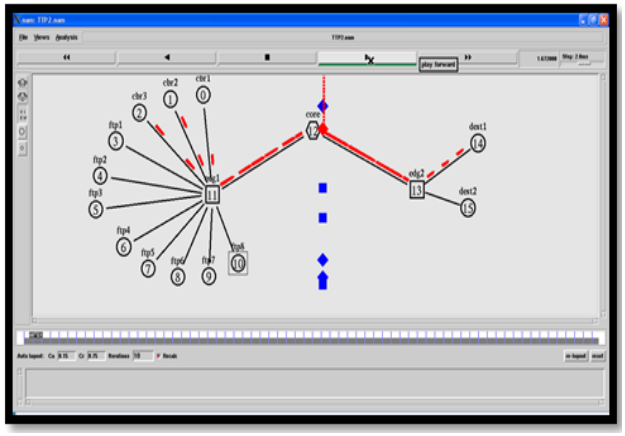

Fig. 13. Simulation topology for scenario (2)

Table III and Table IV present summery of second simulation scenario in comparison with three type of polices. Table III shows the simulation results for CBR traffic, where Table IV displays results for FTP packet.

TABLE III: DROPPED CBR PACKETS IN SCENIRO2 WITH THREE TYPES OF POLICY

Policy type	Packet gen	Packet size	Sent packet	Forwarded packet	Dropped packet	Packet lost	Packet loss rate	Packet loss rate %	Average delay
Token bucket	79251	1000	79251	134777	50374	50438	0.6356	64%	0.09978238094
TP	79251	1000	79251	178394	29619	29682	0.3737	37%	0.09908690056
TSW2CM	79251	1000	79251	177752	29943	30003	0.3778	38%	0.08895992846

TABLE IV: DROPPED FTP PACKETS IN SCENIRO2 WITH THREE TYPE POLICIES

Policy type	Packet gen	Packet size	Sent packet	Forwarded packet	Dropped packet	Packet lost	Packet loss rate	Packet loss rate %	Average delay
Token bucket	99480	40	99480	195460	50374	50442	0.507057	51%	0.06694330466
TP	79511	40	79511	179174	29619	29682	0.373307	37%	0.08872725132
TSW2CM	79860	40	79860	179579	29943	30003	0.375695	38%	0.08811954247

3) Third scenario

Fig. 14 shows the topology for the third scenario. In this scenario, more source nodes are added for more accurate results and some nodes such as PC1 and PC2 used to generate TCP traffic over UDP. Four destination nodes are added to increase the number of nodes in order

to reduce the incidence of congestion. NAM runs during 40.399213 seconds of simulation and process time.

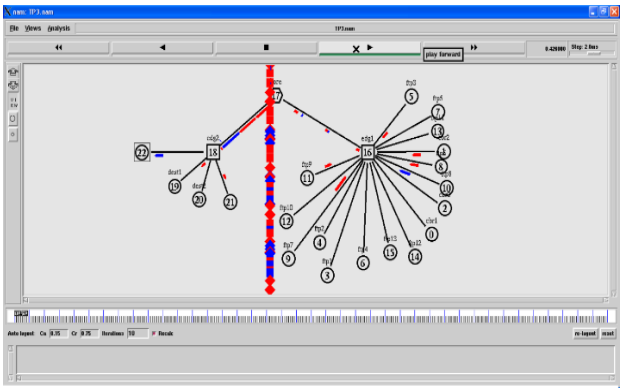

Fig. 14. Simulation of the third scenario

TABLE V: DROPPED CBR PACKETS IN SCENIRO 3 WITH THREE TYPE POLICIES

Policy type	Packet gen	Packet size	Sent packet	Forwarded packet	Dropped packet	Packet lost	Packet loss rate	Packet loss rate %	Average delay
Token bucket	59705	1000	59705	107513	35733	35802	0.598493	65%	0.08101830597
TP	59705	1000	59705	109445	34770	34836	0.58236	37%	0.08870621553
TSW2CM	59705	1000	59705	109339	34822	34889	0.583234	38%	0.08822580017

B. *Results Analysis*

In this section, the analysis of main results will focus on the drop packets and the average Delay, to show how QoS achieved.

As can be noted from Table VI, clear reduction in the number of dropped packets during the run compared to the three types of police, 29,619 is the number of dropped packets, which means high reducing rate of lost packets with Token Packet 64%, 38% with TSW2CM, the lowest rate with TP policy is 37%. This low percentage in the loss ratio ensures applications protection time during when network congestion occurs, and ensures the protection of ACK packet.

TABLE VI: SUMMARY OF SIMULATION RESULTS FOR ALL SCENARIOS (CBR TRAFFIC)

Scenario	Drop CBR packets in three policies		
	TP	**Token bucket**	**TSW2CM**
First	**0**	0.0039	0.003947
Second	**0.37374**	0.6356	0.3778
Third	0.58236	0.598493	0.583234

V. CONCLUSION

In this article we presented an enhanced RED algorithm in order to minimize packet loss of sensitive traffic flows. The modified RED (MRED) discards the packets from the queue when the buffer becomes congested. In contrast to RED, MRED adds two virtual queues to isolated mixed traffic. One for UDP packet and the other one for TCP packet. MRED used a new policy called TP policy to mark every packet with different code point based on type of protocol.

The model is implemented using NS-2 simulator. Three scenarios are used to evaluate the model and compare it with two types of policies, i.e. Token Becket and TSW2CM. The results have showed that MRED achieves low loss rate compared to the RED and the two other policies, i.e. token bucket and TSW2CM.

The biggest challenge facing the new proposal is how to reduce the percentage of average delay especially with heavy loud applications, such as video conferencing and voice applications. This issue will be our near future focus.

REFERENCES

[1] C. V. Hollot, V. Misra, D. Towsle, and Wei-Bo Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proc. IEEE/INFOCOM*, Anchorage, Alaska, April 2001.

[2] M. Allman, V. Paxson, and E. Blanton, "TCP congestion control, request for comments," *RFC 2581 Internet Engineering Task Force*, September 2009.

[3] U. Bodin, O. Shelen, and S. Pink, "Load-tolerant differentiation with active queue management," *ACM SIGCOMM Computer Communication Review,* vol. 30, no. 3, pp. 4-16, July 2000.

[4] Internet World Stats Web Site. [Online]. Available: http://www.internetworldstats.com/stats.htm

[5] K. Ramakrishnan and R. Jain, "A binary feedback scheme for congestion avoidance in computer networks," *ACM Transactions on Computer Systems*, vol. 8, no. 2, pp. 158-181, 1990.

[6] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transacton on Networking*, vol. 1, no. 4, 1993.

[7] F. Sally and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM*

Transactions on Networking, vol. 1, no. 4, pp. 397-413, 1993.
[8] RFC 2309, Recommendations on Queue Management and Congestion Avoidance in the Internet. (April 1998). [Online]. Available: http://www.rfc-editor.org/rfc/rfc2309.txt
[9] S. Madipelli, "The RED algorithm–averaged queue weight modeling for non linear traffic," Ph.D. thesis, Blekinge Institute of Technology, Karlkrona, Swede, 2009.
[10] J. Chung and M. Calypool, "Analysis of active queue management computer science department," in *Proc. Second IEEE International Symposium on Network Computing and Applications*, Cambridge, Massachusetts, April 16-18, 2003.
[11] H. Wang and K. G. Shin, "Refined design of random early detection gateways," Department of Electrical Engineering and Computer Science, Real Time Computing Laboratory, University of Michigan.
[12] The Network Simulator - NS-2. [Online]. Available: http://www.isi.edu/nsnam/ns/
[13] E. Weingartner, H. vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," in *Proc. IEEE International Conference on Communications*, Dresden, Germany, June 2009.

Adel Smeda is holding a Ph. D. degree in computer science from University of Nantes, France, a master degree in computer engineering from Dalhousie University, Canada, and B.S. degree in computer engineering from Bright Star University of Technology, Briga, Libya. His area of interest includes computer networks, modeling, and software architecture. Right now he works as an associate professor at the University of Al-jabal Al-gharbi, Libye.

El-Bahlul Fgee received his PhD in Internetworking Department of Engineering Mathematics and Internetworking, Dalhousie University Halifax NS, Canada in 2006. His is working in the area of Networking and Security and published papers in International Journals and International Conferences. He is teaching and supervising MSc students at the Libyan Academy, Tripoli-Libya. He worked as the Dean of Gharyan High Institute of Vocational Studies from 2008 -2012.

Khadeja Abu Algassm graduated from the Libya Academy, Department of Electrical and Computer Engineering in 2012. She worked in the area of Networking Quality of Service (QoS). She is recently teaching undergraduate students.