International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No.5 (Sep-2020)

http://dx.doi.org/10.12785/ijcds/090521

Reconfigurable Nonlinear GRED Algorithm

Ahmad F. AL-Allaf[1] and A. I. A. Jabbar[2]

[1]Dept. of Computer Eng. ,Engineering Technical college-Mosul, Northern Technical University, Mosul, IRAQ
[2]Dept. of Electrical Eng., College of Engineering, University of MOSUL, Mosul, IRAQ

Received 11 Dec. 2019, Revised 25 Jul. 2020, Accepted 26 Aug. 2020, Published 1 Sep. 2020

Abstract: In this paper, a reconfigurable queuing scheme based on Gentle Random Early Dropping (GRED) algorithm has been proposed. Two nonlinear GRED algorithms have been suggested to study the effect of maximum threshold (max_{th}) and maximum dropping probability (max_p) in the GRED algorithm. In both algorithms, the nonlinearity degree of the two parts of the dropping probability curve of the GRED algorithm is computed to prevent the discontinuity between them.
After that, a third algorithm (called Reconfigurable Nonlinear GRED "RNLGRED") is proposed. This algorithm uses the reconfigurability technique to select the best nonlinearity configuration between the first two proposed algorithms. A simulation carried out in OPNET shows that the proposed RNLGRED algorithm achieves a significant reduction in the queuing delay, jitter, and average queue size when compared with the classical GRED scheme. The reduction in queuing delay, jitter, and average queue size, for example in scenario-1, is up to 40%, 40%, and 35% respectively according to the implemented scenario.

Keyword: GRED Algorithm, Queuing Schemes, Reconfigurable Queuing Algorithm

1. INTRODUCTION

Active Queue Management (AQM) algorithms can reduce the number of packets dropped by routers and provide greater capacity to absorb traffic bursts without dropping packets while reducing end to end packets delays [1]. Random Early Detection (RED) queueing algorithm is one of the early methods of AQM that was proposed in the early 1990s by Floyd and Jacobson[2] and was subsequently adopted by the Internet Engineering Task Force (IETF). The RED mechanism was originally developed to work in conjunction with transport-layer congestion control protocols such as TCP. However, X. Wang [3] concludes in his thesis, that RED is also able to control the queuing delay and jitter of UDP based traffic under different load conditions and can provide a good solution to quality degradation of real-time traffic under congested network.

However, Nicolas K. et al.[4] mention in their paper, that there is no single overall best AQM scheme, as each scheme proposes a specific trade-off. In recent years, two new AQM methods (Proportional Integral controller Enhanced (PIE) [5,6] and Controlled Delay (CoDel) [7,8]) have been developed by the IETF AQM working group to solve the full buffer problem "bufferbloat" in the network by reducing the queuing delay in routers. Both algorithms try to keep configuration parameters to a minimum, self-tune, and control queue delay around a target value. Also, both exhibit high latency during transient congestion periods.

Nonetheless, improving already existing AQM algorithms such as RED and GRED[9] are better than start on a complete new AQM reimplementation[10,11], as recommended by European Commission recommended in its project named "RITE" (Reducing Internet Transport Latency)[12]. Since the reimplementation of a new algorithm can involve considerable development cost and effort. Also, the advantages of developing existing AQM can be made available quickly to all existing users, without having to wait for the next round of hardware purchases[12]. However, there is a small performance difference between different AQMs. Therefore using any AQM will give a benefit compared to no AQM.

The current paper adopts this point of view and suggests a modified improved version of the existing AQM algorithm (GRED algorithm) using nonlinearity and reconfigurability techniques. Although RED algorithm has distinct advantages, also it has many disadvantages. One of the RED problems is the difficulty of predicting the queuing delay due to variation in the queue size with the level of congestion or traffic load. This variation caused by the sudden change in the packet dropping which may have

E-mail address: Ahmadfalih64@gmail.com, Aiajabbar1@yahoo.com

negative effects on network design. To solve this problem, GRED suggested replacing the discontinuity (sharp change) of the packet dropping probability from max_p to 1 by a gentle slope to increase the stability of the RED algorithm.

This paper investigates the influence of using nonlinearity in the dropping probability function of the GRED algorithm, to obtain more smoothly dropping probability curve and study the influence of varying max_p and max_{th} parameters to improve the performance of the algorithm. Furthermore, the reconfigurability technique based on current queuing delay is introduced to select the best nonlinearity configuration with changing of traffic load. The queueing delay has been concentrated rather than propagation delay because in many cases queueing delay is the most important component of end-to-end delay over the internet [13].

The rest paper of the paper is organized as follows: The next section gives an overview of the proposed RNLGRED algorithms. Section 3 details the proposed three algorithms. The reconfiguration process is described in section 4. Section 5 discusses the simulation results and the conclusion is given in section 6.

2. The Proposed Architecture of the RNLGRED Model

The proposed RNLGRED model consists of two parts:

- Part-1: In this part, two modified nonlinear GRED algorithms are proposed to study the effect of using different degrees of nonlinearity in region 2 and region 3 of the GRED algorithm (as shown in Fig. 1) on the performance metrics (queuing delay, jitter, average queue size (avgQ), and packet drop). These metrics are built-in statistics in OPNET software. The influence of the varying the parameters max_p and max_{th} as they determine the intensity of the congestion control is also investigated. The two proposed algorithms are:

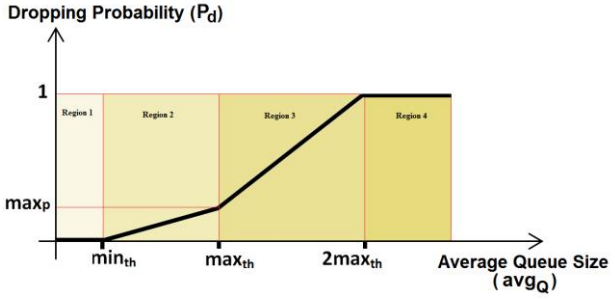

Figure 1. Drop/mark probability of the GRED

a) **NLGRED_1**: In this algorithm, the max_{th} parameter is fixed with changing max_p parameter at different degrees of nonlinearity.

b) **NLGRED_2**: In this algorithm, the max_{th} parameter changed with fixing max_{th} parameter at different degrees of nonlinearity.

In both cases, the nonlinearity degree for GRED dropping probability in region 2 and 3 is derived, so that there is no discontinuity between the two dropping probability (P_d) curves.

- Part-2: In this part, a third algorithm "called RNLGRED" is proposed using the reconfigurability technique to switching between the best nonlinearity configuration of the previous two algorithms; (NLGRED_1 and NLGRED_2), to obtain the best result with the changing of the traffic load based on the packet queuing delay.

3. Non-Linear GRED (NLGRED) Algorithm

GRED adopts a linear packet dropping probability (P_d) that tends to be less aggressive at light load and not aggressive enough at high load when the average queue size approaches the max_{th}. This behavior is unacceptable for the dropping mechanism at the router. So an improved GRED algorithm is proposed using a nonlinearity scheme to make the packet dropping function more rational and variability smoother. Two types of improvements in the GRED algorithm are investigated based on using nonlinearity in packet dropping function:

A. Nonlinear GRED with fixed max_{th} and variable max_p (NLGRED_1)

In this case, the degree of nonlinearity (k) of P_d curve in region 2 of Fig. 1 (when $min_{th} \le avgQ < max_{th}$) is changed in steps (1, 1.5, 2, 3, 4 and 5)). In each step, the non-linearity degree (j) of the P_d curve in region 3 ($max_{th} \le avgQ < 2max_{th}$) is calculated so that there is no discontinuity between the two curves of P_d in region 2 and 3. The value of max_{th} is constant in all these cases while the max_p is changed according to the point of convergence of the two curves. This algorithm is named as NLGRED_1. Table 1, shows the values obtained using (1) and (2) in the MATLAB program.

$$P_d=max_p*((avgQ\text{^}k-min_{th}\text{^}k)/(max_{th}-min_{th})\text{^}k) \quad min_{th} \ge avgQ \ge max_{th} \quad (1)$$

$$p_d=(((1-max_p)*(avgQ-max_{th}))/max_{th}+ max_p)\text{^}j \quad max_{th} \ge avgQ \ge 2*max_{th} \quad (2)$$

Fig. 2 shows a P_d function plot with the avgQ of using the values of nonlinearity degree in parts 2 and 3 of the GRED dropping probability function with constant max_{th}, as illustrated in Table 1 using MATLAB program. As shown in Fig. 2, to maintain a constant value of max_{th} at 300, the value of max_p increased with the increasing value of k.

TABLE 1 NONLINEARITY DEGREE IN NLGRED_1

k	j	max_p
1	1	0.1
1.5	0.8309	0.1475
2	0.7008	0.1985
3	0.4894	0.3228
4	0.3018	0.4962
5	0.1217	0.7497

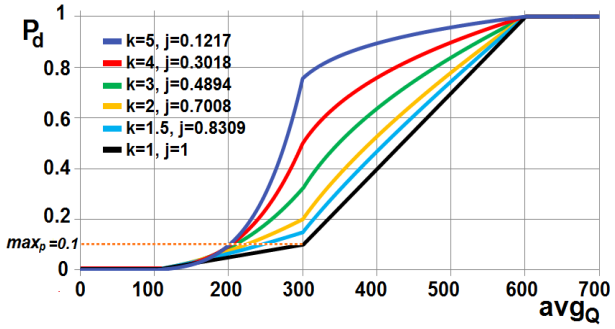

Figure 2. Dropping probability (P_d) with the avgQ in NLGRED_1

B. *Nonlinear GRED with variable max_{th} and fixed max_p (NLGRED_2)*

The performance of GRED depends, partially, on the thresholds. If the thresholds are small, then the link utilization will be low. On the other side, if the thresholds are set too high, the congestion might occur before notifying the receiver. The optimal values for min_{th} and max_{th} depend on the desired avgQ. Also, the end-to-end delay depends on the max_{th}; the min_{th} has a small impact on the end-to-end delay distribution. Besides, large max_{th} results in a reduced loss rate [3].

In this second proposed algorithm, the max_{th} of the router queue is changed by multiplying max_{th} by a factor "m", which takes values from 0.6 to 1.2 in steps equal to 0.2. In each case, the value of the non-linearity degree is computed for P_d in region 2 (i.e. at $min_{th} \le avgQ < max_{th}$) and in region 3 (i.e. at $max_{th} \le avgQ < 2max_{th}$) of the P_d curve so that there is no discontinuity between the two curves. The value of max_p is constant at 0.1 and P_d at the meeting point of the between the two parts of the curve is constant at 0.2 in all these cases. The value of P_d =0.2 represents the value of nonlinear P_d at max_p=0.1, avgQ=m*max_{th} (and k as in Table 2) in (3). This value of P_d=0.2 is then used to calculate the nonlinearity degree (j) in region 3 of P_d curve using (4) so that there is no discontinuity between the curves of two parts of P_d. This algorithm is named as NLGRED_2. Table 2, shows the values obtained using (3) and (4) in the MATLAB.

TABLE 2 NONLINEARITY DEGREE IN NLGRED_2

m	k	j
0.6	1.5093	0.7020
0.8	1.74151	0.7012
1	2	0.7008
1.2	2.2966	0.7005

$$P_d = max_p * ((avgQ \hat{} k - min_{th} \hat{} k)/(m*max_{th} - min_{th}) \hat{} k)$$
$$min_{th} \le avgQ < max_{th} \quad (3)$$

$$P_d = (((1-max_p)*(avgQ - m*max_{th}))/((2-m)*max_{th}) + max_p) \hat{} j$$
$$max_{th} \le avgQ < 2*max_{th} \quad (4)$$

Fig. 3 shows a P_d function plot with the avgQ of using the values of nonlinearity degree on parts 2 and 3 of the GRED dropping probability function, with constant max_p=0.2 as illustrated in Table 2 using MATLAB program. Note that avgQ in (3) and (4) is equal to m*max_{th}. According to Fig. 3, when the average queue size is small (less than max_{th}), the increase of corresponding packet-dropping probability is little. When the avgQ approach to the min_{th}, the packet dropping probability speed decreases so that as many packets to services. When the avgQ is greater than max_{th}, then packet dropping probability increased to increase the number of dropped packets.

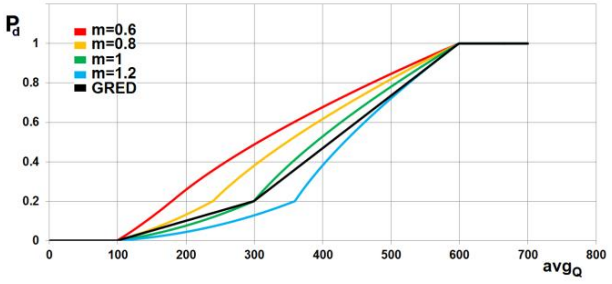

Figure 3. Dropping probability (P_d) with the avgQ in NLGRED_2

However, the value of max_{th} should not be chosen too large to avoid a high queuing delay as the average queue size could excessively exceed its target value. Finally, as shown from Figs. 2 and 3, that the probability of packet loss by using the nonlinear scheme is relatively smooth which stabilizes the network performance comparing to suddenly change or discontinuity in the probability of packet loss.

C. *Reconfigurable nonlinear GRED (RNLGRED) algorithm*

In the second part of the proposed algorithm, the reconfigurability is used to switching between the best nonlinearity configuration of the two described algorithms (NLGRED_1 and NLGRED_2). Fig. 4 illustrates the block diagram for the architecture of the proposed system.

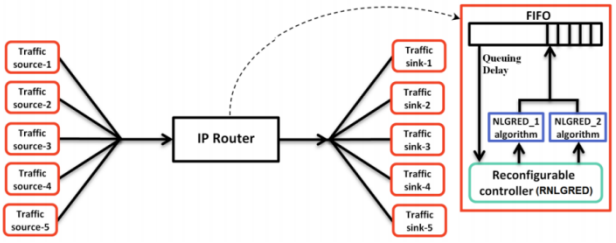

Figure 4. The Network architecture of the proposed RNLGRED system

The Reconfigurable controller is implemented RNLGRED algorithm. It used to select the AQM algorithm (NLGRED_1 or NLGRED-2) that gives the best queuing delay. The selected AQM algorithm will affect both the QoS offered to packets passing through the network and the overall performance (throughput, delay, delay jitter, and fairness) of the network.

4. Reconfiguration Process

The aim of using the reconfiguration in the proposed system is to minimize the queue delay which is a major component of the QoS delivered to the users. Network operators would normally want to get a rough estimate of average delays in their congested routers. To achieve such predictable average delays with GRED, it would require constant tuning of the parameters to adjust the current traffic conditions.

The delay threshold is assigned automatically. Fig. 5 presents a flowchart for this approach which represents the primary task of the RNLGRED algorithm. Three parameters are used, *m*, *w*, and *x* in the reconfiguration procedure to select the delay threshold automatically. The first parameter (*m*) determines the length of the test period which is initialized at current simulation time and then increment 5 seconds each period. This means that the average queuing delay of the active queuing algorithm is tested every 5 seconds. However, this value depends on the oscillation in the delay parameter. If the oscillation is high, a small increment value for *m* (e.g. 1 or 2 seconds) should be selected and it can be increased if the oscillation in the delay is low.

The second parameter (*w*) is equal to the number of *m* periods. This *w* period is used to test the queuing delay under the second inactive queuing algorithm. In other words, after every *w* period (*w*=15 m in this scenario), this procedure forces the reconfiguration system to switch to the alternate (inactive) AQM algorithm for one second period to perform a quick test for the queuing delay under the alternate (inactive) AQM. Based on the queuing delay comparison in active and inactive AQM, the reconfiguration controller decides if stay on the active AQM or switching to the inactive AQM.

Finally, parameter x is used by RNLGRED algorithm to select NLGRED_1 or NLGRED_2. At the Initialization: m= current simulation time, w=0, and x=0. The values of the parameters m and w are placed by the network operator and depend on the monitoring of the change in the queuing delay and the experience of the operator.

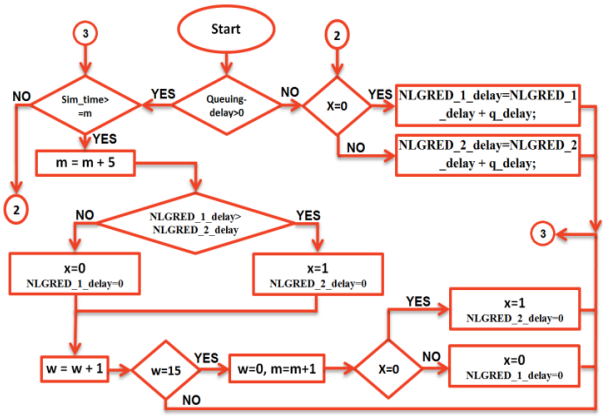

Figure 5. Flowchart of the reconfiguration process used by the RNLGRED algorithm

5. Simulation results of Reconfigurable Nonlinear GRED (RNLGRED) algorithm

This section provides the details of simulating the proposed RNLGRED model using OPNET modeler 14.5 software.

A) RNLGRED model assumptions

The following are the assumptions used in the simulating the proposed RNLGRED model:

1- The QoS scheme applied across the router interface is FIFO.

2- The network topology used is a dumbbell[14] with a single bottleneck as in Fig. 6. There are 5 traffic sources and five traffic sinks sharing two routers.

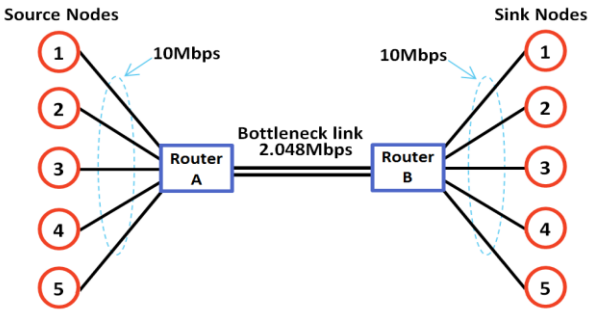

Figure 6. Network topology of the simulated scenarios

3- Three scenarios are implemented. The traffic load in the first scenario is only UDP traffic, in the second scenario is mixed from UDP and TCP, and in the third scenario is TCP only. The characteristics of the traffic sources in scenario_1, scenario_2, and scenario_3 are as indicated in Table 3, Table 4, and Table 5, respectively.
4- The link between the two routers is the bottleneck link with a capacity of 2.048Mbps (PPP_E1) in scenario-1 and 1.544Mbps (PPP_DS1) in scenario-2 and scenario-3
5- The connection bandwidth between each traffic source (or sink) and the router have a capacity of 10Mbps.
6- Two proposed queuing disciplines, NLGRED_1 and NLGRED_2, are used alternatively to serve the single FIFO queuing system and to support the proposed reconfigurable system.
7- The third proposed reconfigurable AQM discipline performs the reconfiguration by switching between the NLGRED_1 and NLGRED_2 AQM algorithms based on the value of the queuing delay of the FIFO queue.
8- In this simulations, the loads in the traffic sources are activated at different simulation times to study the performance of the proposed AQM algorithm under different traffic load and thus different levels of congestion on the bottleneck link.
9- The parameters of NLGRED_1, and NLGRED_2 in both scenarios are set as min_{th}=100, max_{th}=3*min_{th}, w_q=0.002, and max_p=0.1 as recommended in[9]. The procedure of the reconfiguration process (shown in Fig. 3) is implemented in the oms.qm source file of the Queue Management sub-package in the Opnet Model Support (OMS) package.

TABLE 3. CHARACTERISTICS OF THE TRAFFIC SOURCES IN SCENARIO 1

Traffic source No.	**1**	**2**	**3**	**4**	**5**
Frame/sec	30 frame/sec	30 frame/sec	30 frame/sec	30 frame/sec	30 frame/sec
Frame size	6554 byte/frame	2185 byte/frame	874 byte/frame	3495 byte/frame	4369 byte/frame
Starting time(sec)	100	300	400	500	600
Ending time(sec)	1000	1000	1000	750	750
Type of service	streaming traffic	excellent effort	Standard	background	streaming traffic
Traffic in bit/sec	1572960b/s	524400b/s	209760b/s	838800b/s	1048560b/s
Accumulated traffic	1572960 b/s	2097360 b/s	2307120 b/s	3145920 b/s	4194480 b/s
%of full load	%75	%100	%110	%150	%200
Region	A	B	C	D	E

TABLE 4. CHARACTERISTICS OF THE TRAFFIC SOURCES IN SCENARIO 2

Traffic source type	**FTP**	**E-mail**	**Voice**	**Video conference_1**	**Video conference_2**
Frame/sec	-	-	-	**10**	**10**
Frame size (bytes)	-	-	-	**20000**	**17280**
Encoder scheme	-	-	**G.711**	-	-
Voice frame/packet	-	-	**1**	-	-
File size (bytes)	10Mbyte	-	-	-	-
E-mail size (bytes)	-	8Mbytes	-	-	-
Type of service	standard	standard	Interactive voice	Streaming Multimedia	Streaming Multimedia
Starting time (sec)	100	200	300	400	600
Ending time (sec)	1000	1000	1000	750	750
Traffic in bit/sec	See Fig. 13				
Accumulated traffic (from Fig. 15)	1525000 b/s	1200000 to 1600000 b/s	1200000 to 1600000 b/s	2000000 b/s	3200000 b/s
%of full load	%98	%78 - %100	%78 - %100	%129	%200
Region	A	B	C	D	E

TABLE 5. CHARACTERISTICS OF THE TRAFFIC SOURCES IN SCENARIO 3

Traffic source type	**FTP1**	**FTP2**	**FTP3**	**FTP4**	**FTP5**
File size (bytes)	20Mbyte	10 Mbyte	10 Mbyte	10 Mbyte	10 Mbyte
Type of service	standard	standard	Standard	standard	standard
Starting time (sec)	100	200	300	400	600
Ending time (sec)	1000	1000	1000	750	750
Received traffic in bit/sec	See Fig. 29				
Total received traffic in bit/sec	See Fig. 29				

10) Since the objective of the simulations is to compare the performances of different queuing algorithms, the absolute values of link speed or buffer size will not affect our results as long as they are set to the same for all algorithms (GRED, NLGRED-1, and NLGRED-2) [15]. However, the 500-packet buffer size was chosen, as a typical buffer size for the access and edge router.

The proposed algorithms are implemented in OPNET modeler 14.5 software and compare its performance with the original GRED algorithm. The algorithms GRED, NLGRED_1, NLGRED_2, and RNLGRED are not from the default AQM algorithms in OPNET as RED and WRED. Therefore, these algorithms has been implemented in the OPNET software as new AQM algorithms which require some modifications in the OPNET's source code. The RNLGRED algorithm aims is to achieve an optimal AQM performance: minimum queue occupancy, least queuing delay and jitter and least packet drop rate. Thus, the analysis of these parameters has been mainly focused.

B) Results and Discussions of Scenario_1

In this scenario, the traffic is only multimedia traffic. Multimedia data transmission over computer networks is sensitive to delay, jitter and packet loss. Fig. 7 shows the traffic sent/received in scenario_1 under three proposed algorithms. The traffic load increasing from %75 to %200 of the full load. In these figures, the regions labeled A-E represent the value of traffic load expressed as a ratio between the generated traffic load (bits/sec) to the capacity of the output link (Full load). Where; A=75%, B=100%, C=110%(light congestion status), D=150%, and E=200% (heavy congestion status).

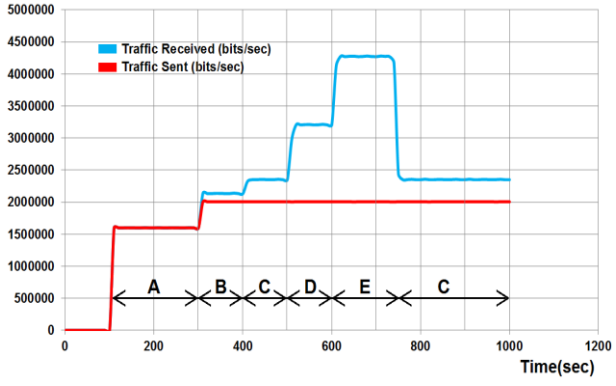

Figure 7. Traffic sent/received in scenario_1 under the three proposed algorithms.

1) Simulation results of NLGRED_1 and NLGRED_2 in scenario_1

The two proposed algorithms (NLGRED_1 and NLGRED_2) are compared using the same performance metrics. Figs. 8 and 9, shows the effect of the implemented algorithms (RNLGRED_1 and RNLGRED_2, respectively) on queuing delay comparing to the original GRED, for the different degrees of nonlinearity under UDP traffic (see Table 3). The GRED curve in Fig. 8 is at k=1 and j=1. As shown in Figs. 8 and 9, the queuing delay is improved using the nonlinearity scheme in two proposed methods.

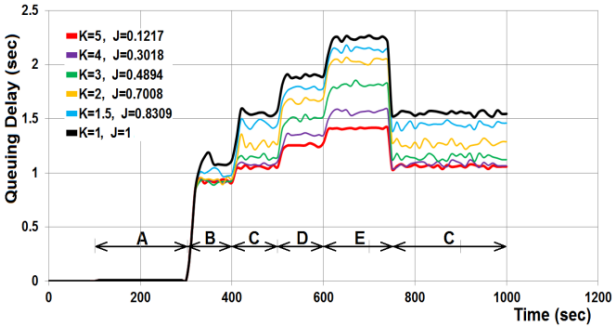

Figure 8. Queuing delay in NLGRED_1 at different nonlinearity degree with the time

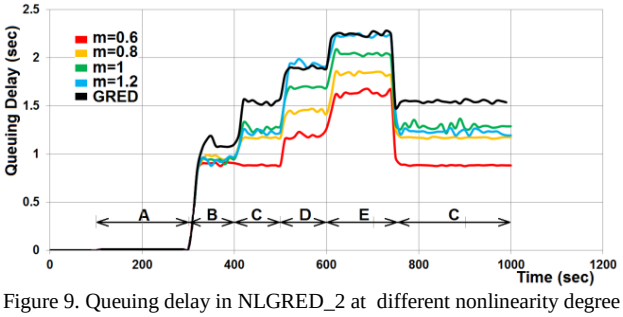

Figure 9. Queuing delay in NLGRED_2 at different nonlinearity degree with the time

Also, Fig. 8 indicates that the values of the delay are lower, as expected, with the increasing max_p settings, while Fig. 9 indicates that the values of the delay are lower, also as expected, for the lower threshold settings (max_{th}). The same things are applied to the average queue size (avgQ), (Figs. 10 and 11) and average jitter (in Figs. 12 and 13).

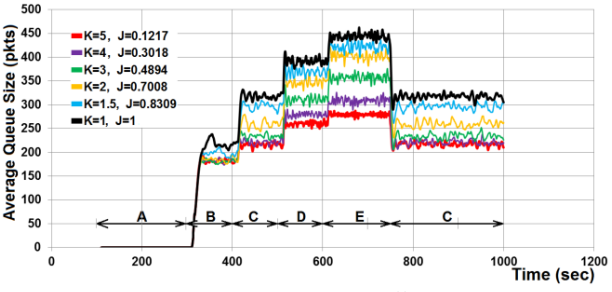

Figure 10. AvgQ in NLGRED_1 at different nonlinearity degree with the time

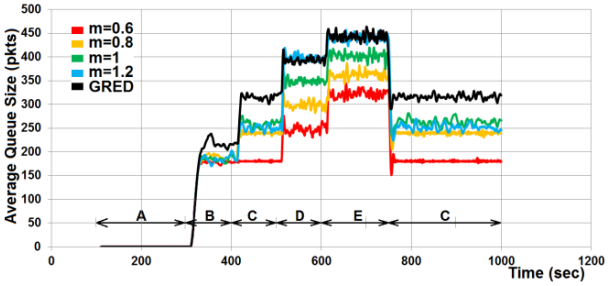

Figure 11. AvgQ in NLGRED_2 at different nonlinearity degree with the time in scenario-1

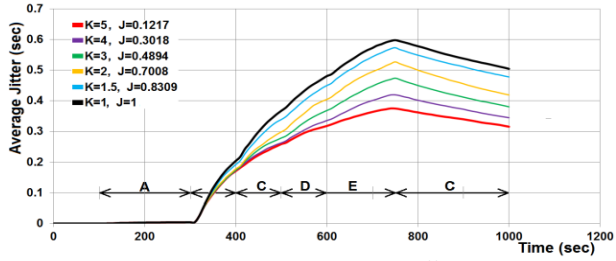

Figure 12. Average jitter in NLGRED_1 at different nonlinearity degree with the time in scenario-1

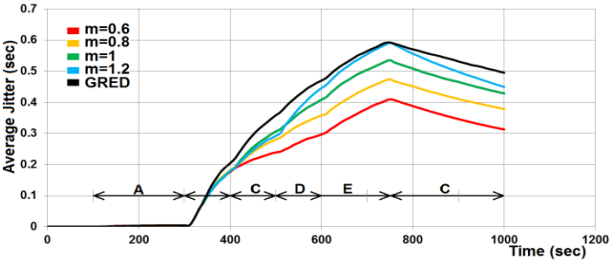

Figure 13. Average jitter in NLGRED_2 at different nonlinearity degree with the time in scenario-1

Traffic dropping shows a slight increase compared to the GRED algorithm in this scenario under the two proposed algorithms (NLGRED-1 and NLGRED-2) as shown in Fig. 14 and Fig. 15. However, the benefits obtained (percentage of decreasing in the queuing delay, jitter, and avgQ), using the proposed algorithms, is much greater than the disadvantages (slight increase in packet dropping).

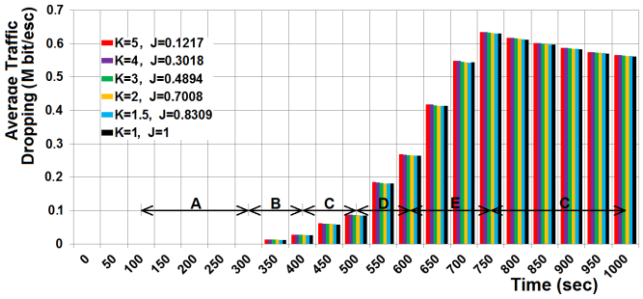

Figure 14. Average traffic dropping in NLGRED_1 at different nonlinearity degree with the time in scenario-1

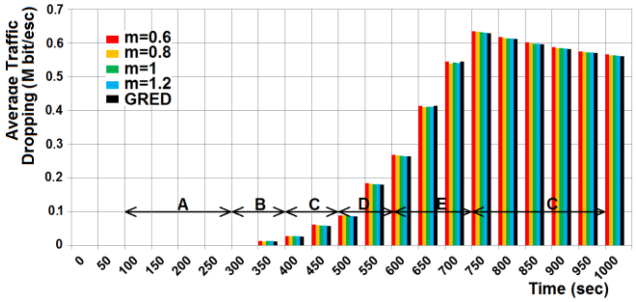

Figure 15. Average traffic dropping in NLGRED_2 at different nonlinearity degree with the time in scenario-1

In the third proposed algorithm, RNLGRED, the best result of queuing delay curves in Figs. 8 and 9 are selected (as well as for jitter, avgQ, and packet dropping) and the reconfigurability technique is used to get the best delay curve between them according to the reconfiguration process described previously. The best result is achieved at k=5 and j=0.1217 in NLGRED_1, and at m=0.6 in NLGRED_2.

2) Simulation results of RNLGRED in scenario_1

Fig. 16 gives the queuing delay curves of three proposed algorithms: in NLGRED_1(at k=5 and j=0.1217), NLGRED_2(at m=0.6), and RNLGRED. The figure shows both the auto-assigned delay threshold (figure a) and the manually assigned delay threshold (figure b). As shown in these figures, the queuing delay in NLGRED_2 outperforms the NLGRED_1 when the traffic load is less than 200% of the full load. However, at heavy congestion status (200% of the full load), the delay in RNLGRED_1 outperform the delay in the RNLGRED_2. The resulting delay curve in RNLGRED follows the minimum delay path between the two delay curves of NLGRED_1 and NLGRED_2.

Fig. 17 shows the average queue size in the RNLGRED algorithm using auto and manually delay threshold assignment. The avgQ should ideally be as low as possible, to ensure low delay. As in the queuing delay in Fig. 16, the avgQ curve in RNLGRED algorithm follows the minimum path between both NLGRED_1 and NLGRED_2. The curve in (a) of RNLGRED contains some ripple due to the test points of the auto-assignment procedure. However, these ripples will be reduced as the variation in the delay is small. Maintaining queue stability is important as some applications that are sensitive to jitter. However, the queue should never be empty, to ensure maximum utilization of the outgoing link.

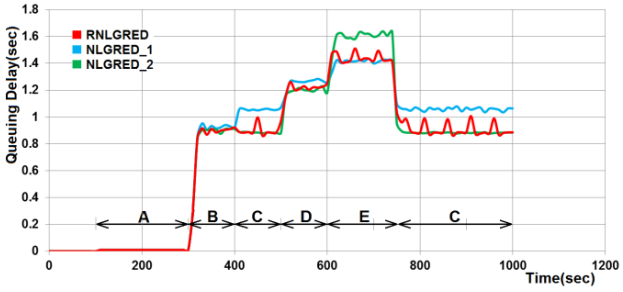

Figure 16. Comparison of queuing delay in NLGRED_1(at k=5 and j=0.1217), NLGRED_2(at m=0.6) and RNLGRED algorithms

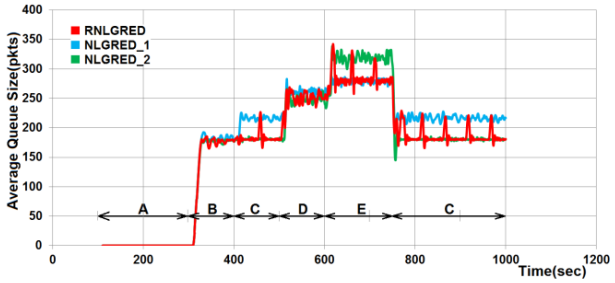

Figure 17. Comparison of average queue size in NLGRED_1(at k=5 and j=0.1217), NLGRED_2(at m=0.6) and RNLGRED algorithms

As in queuing delay and average queue size in Figs. 16 and 17, the average jitter curve of RNLGRED in Fig. 18 trace the minimum path between the two NLGRED_1 and NLGRED_2 algorithms in auto and manually assigned queuing delay.

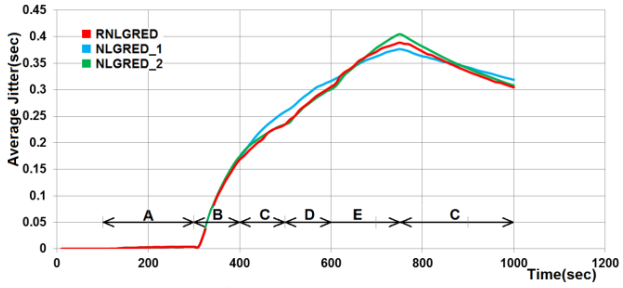

Figure 18. Comparison of average jitter in NLGRED_1(at k=5 and j=0.1217), NLGRED_2(at m=0.6) and RNLGRED algorithms

One can see from simulation results in this scenario, that NLGRED_2 exhibits a more improved behavior than NLGRED_1 in regions C and D, while it exhibits a less improved behavior than NLGRED_1 when the traffic load increased in region E. On the other hand, the NLGRED_1 system exhibits a more stable behavior in E region. This demonstrates that the stability of the two proposed algorithms depends, as in most AQM algorithms, on the traffic load of the system. Since the traffic load of the network is beyond the control of the network manager, therefore it is desirable to have a reconfigurable mechanism (represented by the third proposed algorithm: RNLGRED) whose stability does not depend on the traffic load.

Fig. 19 gives the network average loss in three proposed algorithms under different network loads in scenario 1. Since the traffic load is only UDP traffic (unresponsive traffic), the traffic sources in three algorithms send the traffic at the same rate irrespective of the congestion status. Therefore the link utilization, throughput, and packet drop are the same in the three algorithms.

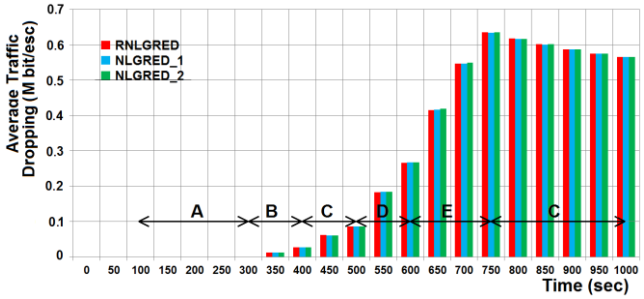

Figure 19. Comparison of average packet dropping in NLGRED_1 (at k=5 and j=0.1217), NLGRED_2(at m=0.6) and RNLGRED algorithms using auto assigned delay threshold

C) Results and Discussions of Scenario-2

In this scenario, the traffic sources consist of a mix of TCP traffic (represented by two data traffic (FTP and email)), and UDP traffic (represented by two videoconferencing traffic and one voice traffic). Audio traffic requires relatively low rates, typically between 5Kbps and 10Kbps for the speech signal and up to 128Kbps for the CD-quality music signal. Bandwidth requirements of video traffic are much greater than those of audio traffic.

A total load of traffic sources is varied according to the competing between TCP and UDP traffic as shown in Fig. 20. The simulations were done on the network topology shown in Fig. 6 and according to the traffic characteristics depicted in Table 4. The bottleneck link in

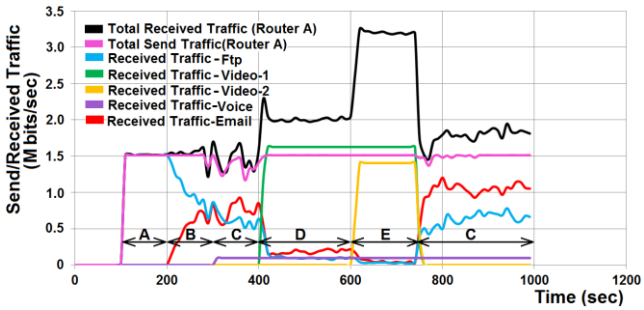

Figure 20. Traffic generated by the traffic sources in the scenario-2 in the three proposed algorithms

this scenario is 1.544Mbps. Other network parameters as in scenario 1.

1) Simulation results of NLGRED_1 and NLGRED_2 in scenario_2

The value of traffic load in the following figures expressed as a ratio between the generating traffic load (bits/sec) to the capacity of the output link (Full load). Where; A=95%, B≈95%, C=100%, D=123% (light congestion status), and E=200% (heavy congestion status).

Figs. 21-26 shows the queuing delay, average queue size, and average jitter under NLGRED-1 and NLGRED-2 algorithms in scenario-2. It is noted from these figures that these statistics show significant fluctuations in regions B and C due to the variation in traffic rate due to the operation of the TCP protocol. However, in regions D and E, UDP traffic is activated and becomes dominant, which stabilizes statistics at a given traffic rate due to traffic rate stability.

In these figures, the proposed algorithms exhibit improved performance compared to the GRED algorithm. The improvement in these statistics increased with the increasing of nonlinearity factor, k in NLGRED-1, and with decreasing the factor, m in NLGRED-2. As in scenario-1, the best result is obtained when k=5, j=0.1217 (NLGRED-1), and at m=0.6 (NLGRED-2).

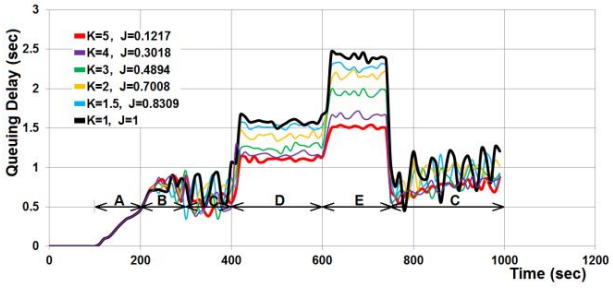

Figure 21. Queuing delay (NLGRED_1) in scenario-2 at different nonlinearity degree with the time

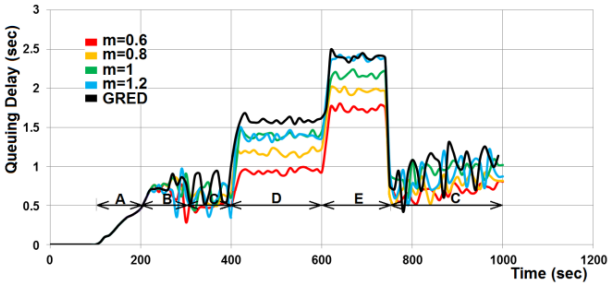

Figure 22. Queuing delay (NLGRED_2) in scenario-2 at different nonlinearity degree with the time

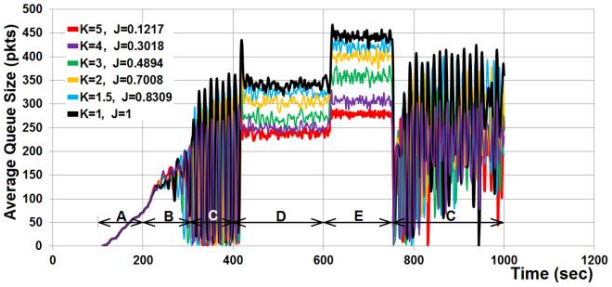

Figure 23. Average queue size (NLGRED_1) in scenario-2 at different nonlinearity degree with the time

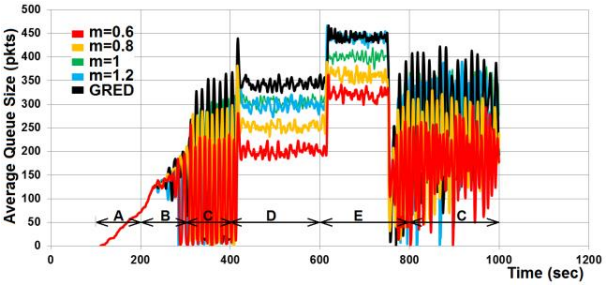

Figure 24. Average queue size (NLGRED_2) in scenario-2 at different nonlinearity degree with the time

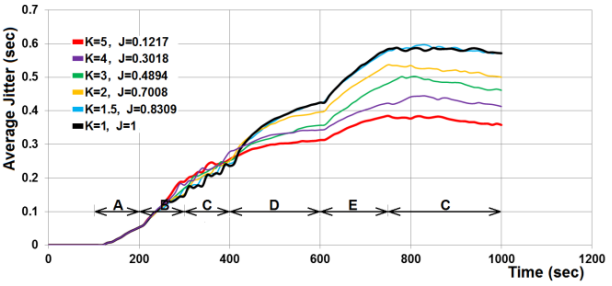

Figure 25. Average jitter (NLGRED_1) in scenario-2 at different nonlinearity degree with the time

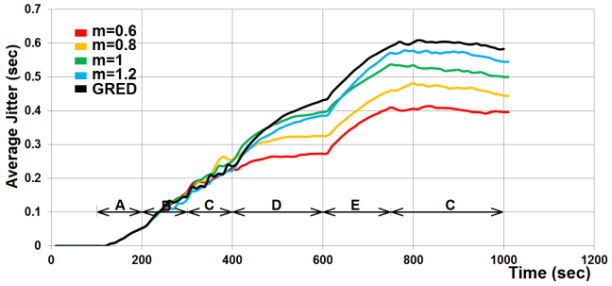

Figure 26. Average jitter (NLGRED_2) in scenario-2 at different nonlinearity degree with the time

Finally, Figs. 27 and 28, gives the average traffic dropping using NLGRED-1 and NLGRED-2 comparing to GRED. The dropping at k=5 (in NLGRED-1) and at m=0.6 (in NLGRED-2) is increased slightly compared to other nonlinearity degrees in NLGRED-1 and NLGRED-2. This is the cost of reducing the queuing delay at this nonlinearity degree comparing to other nonlinearity degrees.

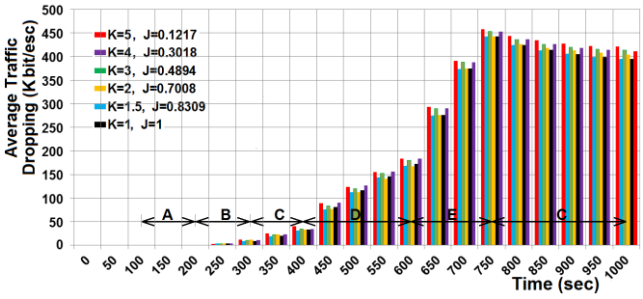

Figure 27. Average traffic dropping (NLGRED_1) in scenario-2 at different nonlinearity degree with the time

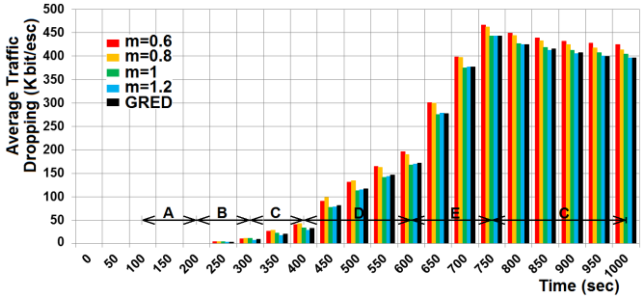

Figure 28. Average traffic dropping (NLGRED_2) in scenario-2 at different nonlinearity degree with the time

2) Simulation results of RNLGRED in scenario_2

Figs. 29-32 gives the corresponding simulation results of the NLGRED_1(at k=5 and j=0.1217), NLGRED_2(at m=0.6) and RNLGRED algorithms using auto delay threshold assignment. In this scenario, the traffic load increasing from %95 to %200 of the full load. In these figures, the regions labeled A-E represent the value of traffic load expressed as a ratio between the generated traffic load (bits/sec) to the capacity of the output link (Full load). Where; A=95%, B≈95%, C=100%, D=123%, and E=200%. The traffic in B and C periods are varied due to the congestion avoidance phase of TCP protocol. When the Email traffic starts growth, the FTP starts to deflation to avoid the congestion.

These figures illustrate the following:: Firstly, the queuing delay in Fig. 29 and avgQ in Fig. 30 of NLGRED_2 is better than NLGRED_1 as the traffic load is less than 200% of the full load. However, in a heavy congestion status (region E), the queuing delay and avgQ in NLGRED_1 is better than NLGRED_2. The same thing is applied to the average jitter as shown in Fig. 31. Secondly, there is a little different for the packet dropping in both algorithms(see Fig. 32) due to maxing responsive and unresponsive traffic in the same FIFO buffer. Thirdly, the competing between TCP flows in region B and between TCP and UDP traffic in regions C-E gives rise to a high oscillation of the average queue size and then queuing delay (see Figs. 29 and 32).

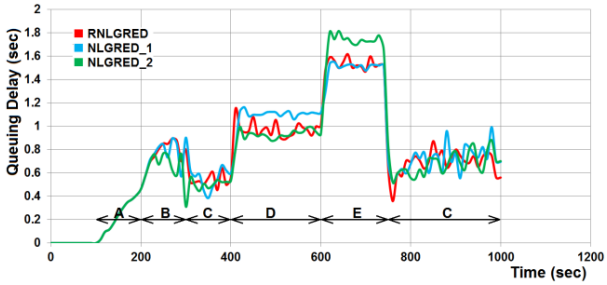

Figure 29. Queuing delay in RNLGRED algorithm compared to NLGRED_1(at k=5 and j=0.1217) and NLGRED_2(at m=0.6) in scenario-2

In all of these figures (Figs. 24-27), the reconfiguration curves of RNLGRED algorithm follow the minimum path to give the best traffic performance. Based on the observation of both scenarios, the following can be explored:

In NLGRED_2 algorithm, the performance of the network is improved using the nonlinearity scheme with decreasing the maximum threshold to 0.6, of its original value, at the light and moderate congestion status. However, at a heavy congestion status (in region E), the increasing of max_p to 0.7497 in NLGRED_1 with the increasing the nonlinearity degree in region 2 of P_d in GRED algorithm up to 5, and at the same time decreasing the nonlinearity degree in region 3 of P_d in the GRED algorithm to 0.1217 gives the better result than NLGRED_2.

Average traffic dropping in Fig. 32 shows a small difference between the three algorithms due to using mixed TCP/UDP traffic comparing to scenario_1 in Fig. 19. The only drawback of existing the TCP and UDP traffic in the same queue is the increased number of TCP packet dropping. When active queue management is used, the number of packets dropped is smaller if only TCP traffic exists. Generally, the presence of UDP traffic causes a state of heavy traffic in the network. Since UDP traffic does not respond to the congestion indicator, more packets must be dropped from TCP traffic to keep the avgQ small.

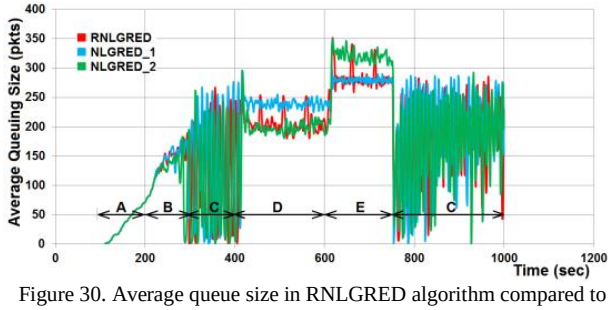

Figure 30. Average queue size in RNLGRED algorithm compared to NLGRED_1(at k=5 and j=0.1217) and NLGRED_2(at m=0.6) in scenario-2

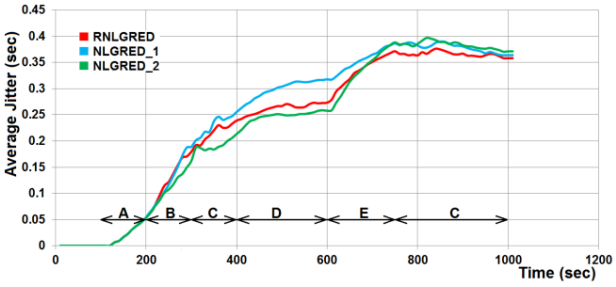

Figure 31. Average jitter in RNLGRED algorithm compared to NLGRED_1(at k=5 and j=0.1217) and NLGRED_2(at m=0.6) in scenario-2

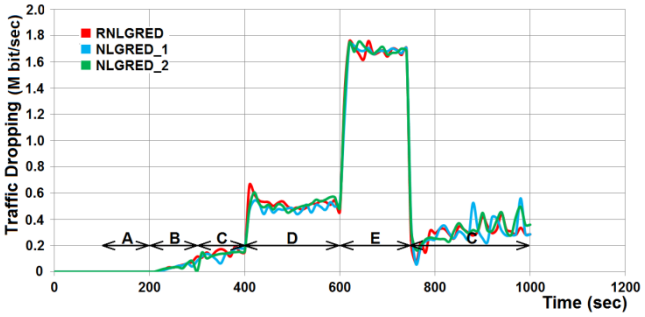

Figure 32. Average traffic dropping in RNLGRED algorithm compared to NLGRED_1(at k=5 and j=0.1217) and NLGRED_2(at m=0.6) in scenario-2

The RNLGRED tracks the best result in both algorithms (NLGRED_1 and NLGRED_2) to take full advantage of both modified algorithms. Although the results of the scenarios show that the proposed algorithms improve the performance of traffic in both UDP network and TCP/UDP network, however, in case of only UDP, network performance is better than mix of TCP and UDP traffic, which means that mixed TCP and UDP network scarify TCP to compensate guaranteed QoS for UDP.

D) Results and Discussions of Scenario-3

In this last scenario, the performance of the proposed algorithms was tested when the traffic is only TCP type. TCP does perform congestion control, but this control creates large fluctuations in the total offered traffic rate in the receiver buffer as shown in Fig. 33. This fluctuation also reflected in the shape of the waveforms of the measured statistics (delay, jitter, drop and avgQ) Therefore, to see the improvement in the performance by using the proposed algorithms, the statistics in this scenario are drawn as average values. Also due to the large fluctuation in the measured statistics in the scenario, manually configuration in RNLGRED is impossible, therefore all RNLGRED statistics are obtained using auto delay assignment.

The simulations were done on the network topology shown in Fig. 6 and according to the traffic characteristics depicted in Table 5. The bottleneck link in this scenario is 1.544Mbps. Other network parameters as in scenario 1 and 2. The regions labeled A-E represent the increase in the traffic load. Where A represents a light congestion status, while and E represents a heavy congestion status.

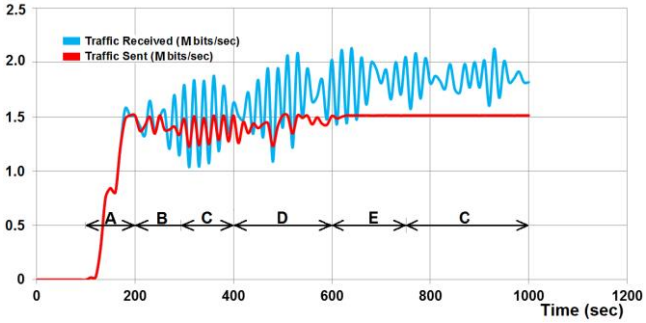

Figure 33. Traffic sent/received in scenario_3 in the three proposed algorithms.

1) Simulation results of NLGRED_1 and NLGRED_2 in scenario_3

Figs. 34-39 show the average queuing delay, average queue size, and average jitter under NLGRED-1 and NLGRED-2 in scenario_3. It is clear in all these figures that there is an improvement in the performance in using the proposed algorithms with respect to the GRED algorithm. It is also noted that this improvement increases with increasing the degree of linearity (k) (increasing max_p) for algorithm NLGRED-1 and with decreasing values of (m) (i.e. decrease of max_{th}) for NLGRED-2. It is noted that at a light congestion status (regions A, B, and C), the best result is obtained at k=2 in NLGRED-1, and at m=0.8 at NLGRED-2. However, with the increasing the congestion status, the best results are obtained when the k increased to 5 in NLGRED-1, and at m reduced to 0.6 at NLGRED-2.

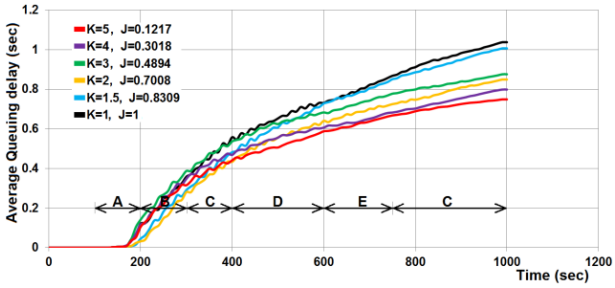

Figure 34. Average queuing delay in NLGRED_1 at different nonlinearity degree with the time in scenario-3

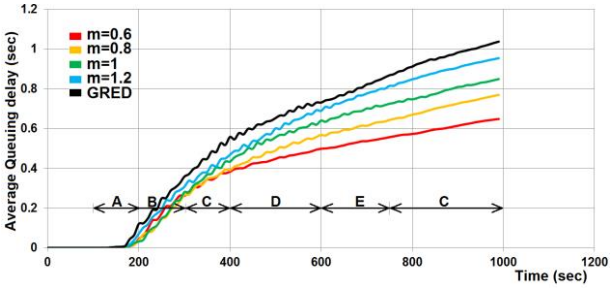

Figure 35. Average queuing delay in NLGRED_2 at different nonlinearity degree with the time in scenario-3

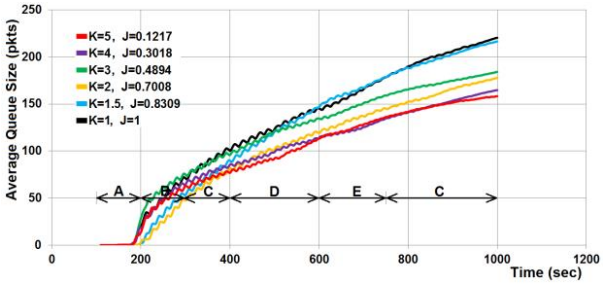

Figure 36. Average queue size in NLGRED_1 at different nonlinearity degree with the time in scenario-3

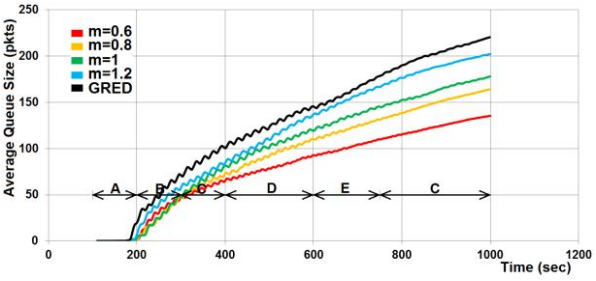

Figure 37. Average queue size in NLGRED_2 at different nonlinearity degree with the time in scenario-3

Average traffic dropping is also increased in NLGRED-1 and NLGRED-2. Comparing with GRED as shown in Figs. 40-41, there is some improvement in packet loss in NLGRED-1 over GRED at k=2 in region C. In NLGRED-2, the improvement is at m=0.8.

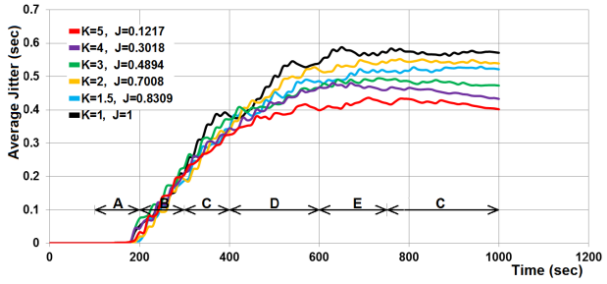

Figure 38. Average jitter in NLGRED_1 at different nonlinearity degree with the time in scenario-3

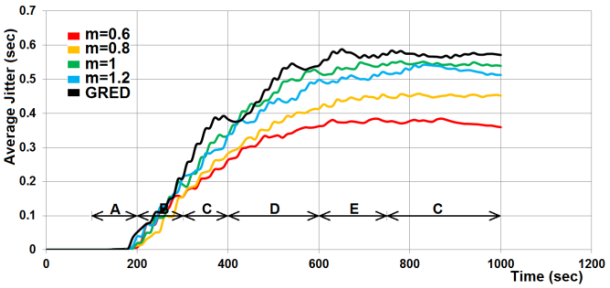

Figure 39. Average jitter in NLGRED_2 at different nonlinearity degree with the time in scenario-3

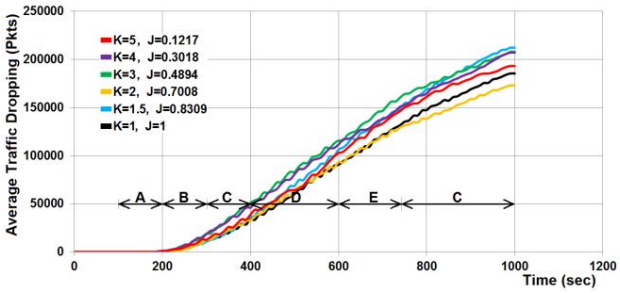

Figure 40. Average traffic dropping in NLGRED_1 at different nonlinearity degree with the time in scenario-3

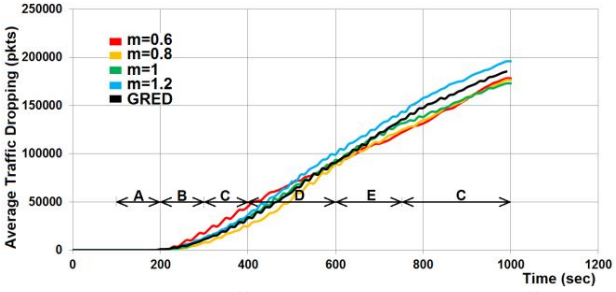

Figure 41. Average traffic dropping in NLGRED_2 at different nonlinearity degree with the time in scenario-3

Overall, the percentage of performance increase is not large according to the percentage of decrease in average delay, jitter, and avgQ. Since the data traffic type is more sensitive to loss of packets than queuing delay and jitter, it is possible to conclude that RNLGRED is more efficient and performs better when the data is only UDP type or UDP and TCP together.

2) Simulation results of RNLGRED in scenario_3

Figs. 42-45 illustrate the average queuing delay, average queue size, and average jitter under the three proposed algorithms: in NLGRED_1 (at k=5 and j=0.1217), NLGRED_2(at m=0.6), and RNLGRED. The figures show that in RNLGRED the values of these statistics follow the minimum path between the two curves of NLGRED_1 and NLGRED_2.

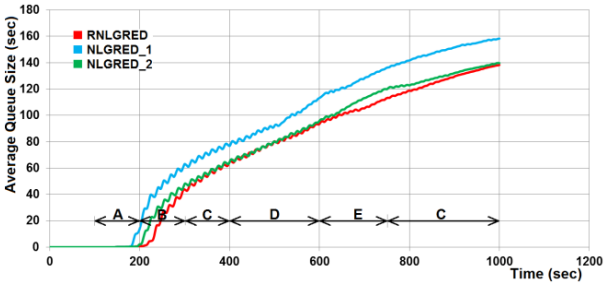

Figure 42. Average queue size in RNLGRED comparing to NLGRED_1, and NLGRED_2 algorithms in scenario-3

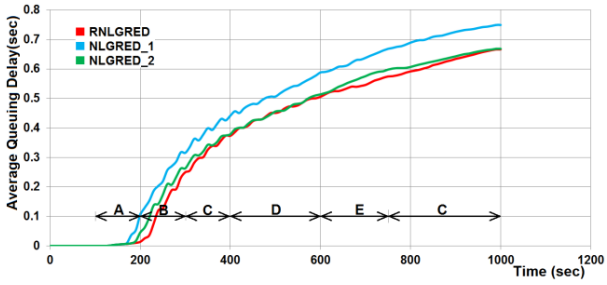

Figure 43. Average queuing delay in RNLGRED comparing to NLGRED_1, and NLGRED_2 algorithms in scenario-3

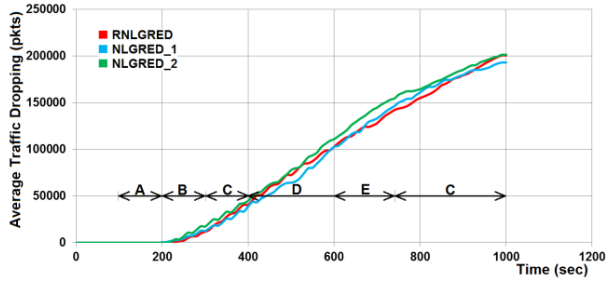

Figure 44. Average traffic dropping in RNLGRED comparing to NLGRED_1, and NLGRED_2 algorithms in scenario-3

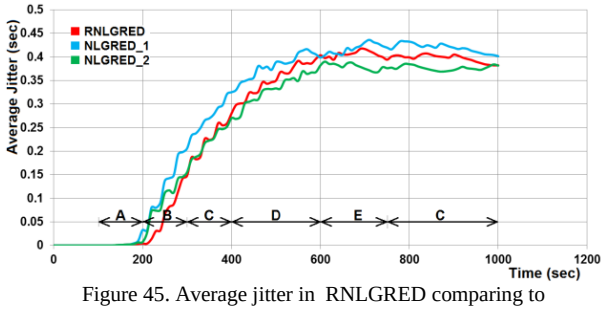

Figure 45. Average jitter in RNLGRED comparing to NLGRED_1, and NLGRED_2 algorithms in scenario-3

6. CONCLUSION

In this paper, developing a packet queuing schemes for the IP routers is presented, to achieve satisfactory throughput and delay performance. The schemes use reconfigurable techniques under different traffic loads and types with a minor modification in the existing IP router queuing algorithms. Two nonlinear GRED-based queuing algorithms (NLGRED_1 and NLGRED_2) are proposed to study the effect of varying max_p and max_{th} parameters in the GRED algorithm.

A simulation carried out in OPNET modular indicates that the proposed NLGRED_1 performs better than NLGRED_2 in heavy congestion status, while NLGRED_2 performs better than NLGRED_1 in light and moderate congestion status.
The third proposed algorithm is a Reconfigurable Nonlinear GRED (RNLGRED), that uses the configurability technique to dynamically select the best nonlinearity configuration between the first two proposed nonlinear GRED algorithms (NLGRED_1 and NLGRED_2).

The RNLGRED algorithm aims to minimize packet delay and jitter, in addition, to stabilize the average queue length around a specific target value. From the comparison of the queuing delay, avgQ and jitter between the GRED algorithm and the RNLGRED for different arrival rate and different traffic types, it is shown that the proposed model has a lower queuing delay, avgQ and jitter (for example in scenario-1, up to 40%, 40%, and 35% respectively), with a small increment in packet dropping rate.

The RNLGRED is tested on a network model with three types of traffic (UDP, mix of TCP and UDP, and TCP). Simulation results show that RNLGRED improves the performance of the system when the traffic is UDP or mixed of UDP and TCP better than TCP only.

REFERENCES

[1] F. Baker "IETF Recommendations Regarding Active Queue Management", Request for Comments (RFC):7567, 2015.

[2] S. Floyd and V. Jacobson. "Random Early Detection Gateways for Congestion Avoidance" IEEE/ACM Transactions on Networking, Vol. 1, No. 4, pp 397–413, 1993.

[3] X. Wang , Active Queue Management for Real-time IP Traffic, Ph.D thesis, Department of Electronic Engineering Queen Mary, University of London, 2006.

[4] K. Nicolas, D. Ros, A. Bagayoko, C. Kulatunga, G. Fairhurst, and N. Khademi "Operating ranges, tunability and performance of CoDel and PIE", Computer Communications, Vol.103, PP.74-82, 2017.

[5] R. Pan, P. Natarajan, C. Piglione, M. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg, "PIE: A lightweight control scheme to address the Bufferbloat

problem," in Proceedings of IEEE 14th International Conference on High Performance Switching and Routing (HPSR), Taipei, Jul. 2013.

[6] R. Pan, P. Natarajan, F. Baker, and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem." RFC 8033 (Experimental), Feb 2017.

[7] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, "Controlled Delay Active Queue Management" Internet Engineering Task Force (IETF), RFC 8289, Jan 2018.

[8] K. Nichols and V. Jacobson, "Controlling Queue Delay," ACM Queue, Vol. 10, No. 5, pp. 20–34, May 2012.

[9] S. Floyd, "Recommendations on using the gentle variant of RED" May 2000, Available at http://www.aciri.org/floyd/red/gentle.html.

[10] A. AL-Allaf and A. Jabbar " Simevents/ Stateflow base Reconfigurable Scheduler in IP Internet Router," International Journal of Computing and Digital Systems (IJCDS), Vol.7, No.5, pp. 291-302, Sep. 2018.

[11] A. AL-Allaf and A. Jabbar "RED with Reconfigurable Maximum Dropping Probability," International Journal of Computing and Digital Systems (IJCDS), Vol.8, No.1, pp. 61-72, Jan. 2019.

[12] P. Hurtig, G. Fairhurst, B. Briscoe, K. Schepper, A. Petlund, N. Khademi, N. Kuhn, and I. Learmonth, RITE: Reducing Internet Transport Latency, European Commission, Project No.317700, November 16, 2015.

[13] T. Tsang , "Performance Modeling and Analysis for Dynamic Bandwidth Distribution Scheme Using Real-Time Probabilistic Systeml, International Journal of Engineering Research and Applications (IJERA), Vol. 3, Issue 4, pp.23142317, Jul-Aug 2013.

[14] S. Floyd, E. Kohler, "Internet research needs better

models", http://www.icir.org/models/, 2002

[15] X. Zeng, C. Lung, and C. Huang, "A Bandwidth-efficient Scheduler for MPLS DiffServ Networks", The IEEE 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2004. (MASCOTS 2004).

Ahmad Falih Al-Allaf received the B.Sc. degree in electrical engineering, M.Sc. degree in electronics and Communication and Ph.D. in computer networks from the department of electrical engineering, University of Mosul,
Iraq. He has published many research papers in the areas of computer networks, fault tolerance computing, reconfigurable computing, and parallel processing. From Nov 2000 to June 2006 he has worked as a lecturer at the dept. of computer science, Benghazi University (Benghazi-Libya). From July 2006 and up to date he is a lecturer with the dept. of computer engineering at Northern Technical University-Technical college/Mosul, Iraq. His research interests include reconfigurable computing, design of the computer network devices, and parallel processing.

A.I.A. Jabbar received the B.Sc. and M.Sc. degree in electrical engineering from Mosul university in 1975 and 1979 respectively, and the Ph.D. degree in communication engineering from the University of Bradford in 1989. His interesting field of research has been concentrated in the area of protocol design for mobile and computer networks. He is now working as a professor in Mosul university and supervising computer and LTE networks Ph.D. projects.