



Active Queue Management for Congestion Control: Performance Evaluation, New Approach, and Comparative Study

Hanaa Mohammed¹, Gamal Attiya² and Samy El-Dolil²

¹Faculty of Engineering, Tanta University, Egypt

²Faculty of Electronic Engineering, Menoufia University, Egypt

Received: 3Jan. 2017, Revised: 20 April. 2017, Accepted: 23 April. 2017, Published: (1 May. 2017)

Abstract: Network congestion is one of the most important problems that effect Quality of Service (QoS). Several Active Queue Management (AQM) algorithms have been developed to avoid congestion problem by controlling the queue length in routers. However, an important problem arising with the current AQM algorithms is that most of the current algorithms handle different traffics by the same strategy. This problem may lead to performance degradation especially for real-time applications ssuch as video and audio traffics. This paper first presents a performance evaluation of the current AQM algorithms. It then presents a new AQM algorithm, called Dynamic Queue RED (DQRED), to guarantee efficient QoS to both real-time and non-real-time traffics. Finally, a comparative study is done between the proposed DQRED algorithm and the most recent AQM algorithms by using the network simulator (NS-2) considering different QoS metrics.

Keywords: Congestion Control, AQM, Dynamic Scheduling, QoS, Real Time Traffics.

1. INTRODUCTION

In the current Internet, network congestion occurs when a network nod carries more data than it can handle. In other words, it occurs at the Internet router when incoming traffic exceeds outgoing bandwidth [1]. This serious problem leads to performance degradation. Typical effects include long queueing delay, multiple packet losses, low link utilization or little useful throughput and poor quality of service [2, 3].

Various congestion control and avoidance mechanisms have been proposed to solve this problem. The proposed mechanisms may be classified into TCP-based congestion avoidance protocols and router-based schemes. The TCP-based protocols are generally called End-to-End congestion avoidance protocols where it monitors packet errors, losses, or delays at the end user to adjust the transmit rate of packets. The router-based schemes are implemented at the internet router. They try to prevent router buffer overflow [4]. The traditional router-based scheme is the First-In First-Out (FIFO) Drop Tail (DT) [5]. This scheme is called passive queue management, as it has no strategy to prevent/avoid

congestion occurrence. It first buffers all incoming packets and then drops any packet arriving after the buffer being full. Hence, several Active Queue Management (AQM) approaches were developed [6-9]. Random Early Detection (RED) is the most popular AQM scheme. It uses an average queue length to drop packets early before buffer overflows. Many RED-based strategies have been developed to rectify numerous problems associated with the RED. These strategies include Adaptive RED, Stabilized RED, Nonlinear RED, Gentle RED, Flow RED, Dynamic RED, etc. Other router-based schemes such as Fair Queuing (FQ), Stochastic Fair Queuing (SFQ) and Deficit Round Robin (DRR), were developed to ensure fair access to network resources and to prevent a bursty flow from consuming more than its fair share.

Although several AQM schemes were developed, they handle all the traffic types by the same strategy. This strategy may lead to performance degradation, especially for real-time traffics as video and audio. This is because different traffics have different QoS requirements. This paper first presents an evaluation of the current AQM schemes by using the Network Simulator NS-2 [10, 11]. Then, it presents a new AQM approach called Dynamic



Queue RED (DQRED), to guarantee efficient QoS to both real-time and non-real-time traffics. Finally, the proposed DQRED algorithm is evaluated and compared with the most recent algorithms.

The rest of this paper is organized as follows. Section 2 introduces the current AQM schemes while Section 3 presents the evaluation results of the current schemes. Section 4 lists the drawbacks of the current schemes while Section 5 describes the proposed DQRED as a new queuing and scheduling strategy. Section 6 presents the simulation results and the performance evaluation of the proposed DQRED strategy compared with the most recent AQM algorithms. Finally, the concluding remarks and the future work are presented in section 7.

2. CURRENT AQM SCHEMES

This section introduces the most recently proposed AQM algorithms for supporting congestion control. These algorithms include Random Early Detection (RED) [12], Virtual Queue (VQ) [13, 14], Fair Queuing (FQ) [15], Stochastic Fair Queuing (SFQ) [16], Random Exponential Marking (REM) [16], Deficit Round Robin (DRR) [17], and Proportional Integrated (PI) [18, 19].

A. Random Early Detection

Random Early Detection (RED) is the most popular AQM strategy. It uses two parameters, minimum threshold (\min_{th}) and maximum threshold (\max_{th}) and calculates the average queue size (q_{avg}). Depending on the calculated value of q_{avg} , the packets may be dropped. As the q_{avg} below \min_{th} , no packets are dropped. If the q_{avg} in the range between the \min_{th} and the \max_{th} , the packets are marked and randomly dropped. While, if the q_{avg} exceeds the \max_{th} , all packets are dropped [12].

B. Adaptive Virtual Queue

Adaptive Virtual Queue (AVQ) is a rate-based technique that provides earlier feedback than normal RED [13, 14]. At each arrival of a packet at the real queue, the virtual queue size is updated. When the virtual queue buffer overflows, the packets are marked or dropped. The virtual capacity of the link is modified such that total flow entering each link achieves a desired utilization of the link. This is done by aggressive marking when the link utilization exceeds the desired utilization and less aggressive when the link utilization is below the desired utilization.

C. Fair Queuing

Fair Queuing (FQ) classifies all arriving packets into different traffic flows and stores each flow into a particular queue [15]. The packets in these queues are served fairly by using the round robin algorithm. FQ assigns a finish time for each packet and provides fair bandwidth allocation, lower delays for responsive sources, and protection from miss-behaved sources. The

complexity of FQ is $O(\log(n))$. where, n is the number of active flows.

D. Stochastic Fair Queuing

Stochastic Fair Queuing (SFQ) provides fair-share of buffer space for each sub-queue. It ensures traffic flow services by hashing and round robin algorithms [16]. A traffic flow may be uniquely identified by four options, *src-address*, *dst-address*, *src-port* and *dst-port*. SFQ uses these parameters to classify packets into one of 1024 possible sub-streams. It does not really allocate a queue for each flow, but it uses a hash function for classification. The available bandwidth is then distributed to all sub-streams based on the round-robin algorithm. The maximum number of packets that can be contained in the whole SFQ queue is 128 packets with 1024 sub-streams.

E. Deficit Round Robin

Deficit Round Robin (DRR) is a scheduling algorithm that handles packets of variable size without knowing their mean sizes [17]. The packet length is determined and a maximum packet size number is subtracted from the packet length. Packets that exceed the number will be held back until the next visit of the scheduler. The DRR serves packets at the head of every non-empty queue whose deficit counter is greater than the packet's size at the head of the queue. If the deficit counter is lower, the queue is skipped (i.e., the packet is not served) and its credit is increased by some given value called *quantum*. The new deficit counter is determined based on the increased value when the scheduler examines this queue for serving its head-of-line packet. After serving the queue, the credit is decremented by value equal to the served packet size. The complexity of DRR algorithm is $O(1)$.

F. Random Exponential Marking

Random Exponential Marking (REM) is an AQM scheme measures the network congestion by a quantity called "price" rather than loss or delay [16]. The REM stabilizes both the input rate around link capacity and the queue length around a small target independent of the number of users sharing the link. The congestion measure price is updated based on the rate mismatch (the difference between input rate and link capacity) and queue mismatch (the difference between queue length and target value). The marking probability depends on the calculated price. Regardless of the number of users, REM attempts to match user rates to network capacity while clearing buffers (or stabilize queues around a small target). In addition, the *end-to-end* marking (or dropping) probability depends on the *sum* of link prices (congestion measures) that summed over all the routers in the path of the user.

G. Proportional Integral

Proportional Integral (PI) is a control-based algorithm with improved stability [18, 19]. It decouples the average queue size from the marking probability $p(t)$ which is in

turn a function of the current instantaneous queue size, the target queue size, and the previous marking probability $p(t - 1)$. Therefore, the PI algorithm uses a combination of two control units: controlling the output queue length, and updating the probability of marking packets.

3. PERFORMANCE EVALUATION OF CURRENT SCHEMES

This section presents a performance evaluation of eight-queue management algorithms, namely, Drop Tail (DT), Random Early Detection (RED), Virtual Queue (VQ), Fair Queuing (FQ), Stochastic Fair Queuing (SFQ), Random Exponential Marking (REM), Deficit Round Robin (DRR), and Proportional Integrated (PI). To investigate the performance of AQM schemes, TCP traffic flows are used in the presence of a Constant Bit Rate (CBR) and multimedia video/audio traffic flows. The CBR and the multimedia applications are considered as unresponsive UDP flows that use a specific amount of bandwidth and do not follow any congestion control strategy. They just send packets blindly at a constant rate. This evaluation is done by using the network simulator NS-2, considering different QoS metrics as throughput, delay, packet loss, and fairness.

Fig. 1 shows a simple network topology used in this evaluation. In this topology, there are two short-lived TCP flows (flow-1 at s1 and flow-2 at s2), one multimedia video UDP traffic at s3, and one unresponsive Constant Bit Rate (CBR) UDP flow at s4. The TCP agent congestion window size is 8000 bytes, the packet size is 1000 bytes with queue limit of 100 packets, the rate for video traffic is 1.5 Mbps with 1000 bytes packet size, and CBR rate is 448 kbps with 210 packet size.

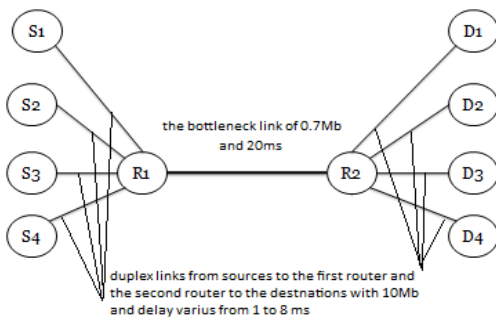


Figure 1. Simple Network Topology

H. Throughput

Throughput is the most widely used performance measure. In computer networks, throughput or network throughput is determined as the number of packets received successfully in a certain amount of time (the simulation time) over a communication channel.

The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot. Throughput is an important factor, which directly indicates the network performance. In addition, throughput of different traffic flows may be used as an indicator to the fairness between different flows.

Fig. 2 shows the throughput of different traffic flows achieved by applying different AQM schemes DT, RED, VQ, FQ, SFQ, DRR, REM and PI. From Fig. 2, FQ, SFQ, and DRR give the same buffer allocation for all traffic types.

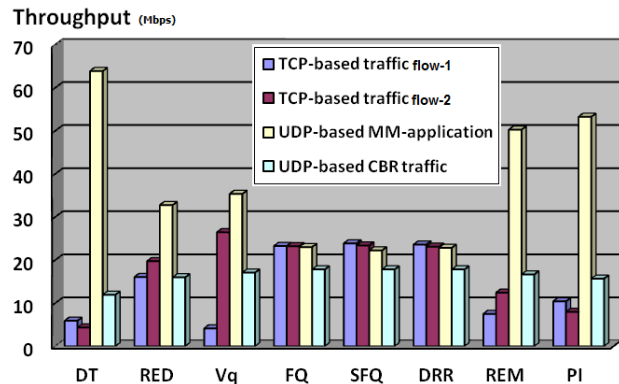


Figure 2. Throughput of different flows at different AQM algorithms

I. Delay or Jitter

Delay is the time taken by a packet to navigate from the source to the destination. It is calculated as the difference between the received time and the transmission time of the packet. Delay is a very important factor of any network because: (i) multimedia applications (audio and video) do not perform well if the delay exceeds a threshold value, (ii) it is difficult to support many real-time applications in very large variations in delay (jitter), (iii) the large value of delay causes difficulty for transport-layer protocols to maintain high bandwidths. The delay can be specified in a number of different ways including; average delay, variance of delay (jitter), and delay bound. Delay jitter is delay variation encountered by packets during transmission over a network.

Fig. 3 shows the delay of different traffic flows that elapsed by applying different AQM schemes DT, RED, VQ, FQ, SFQ, DRR, REM and PI. From Fig. 3, FQ algorithm provides the highest packet delay for UDP-based multimedia traffic flows. With the fair queuing techniques (FQ, SFQ, and DRR), the CBR traffic exhibits the lowest delay. The RED algorithm gives nearly the same delay for all packet types. In addition, it gives the lowest data and video delay compared to other strategies.

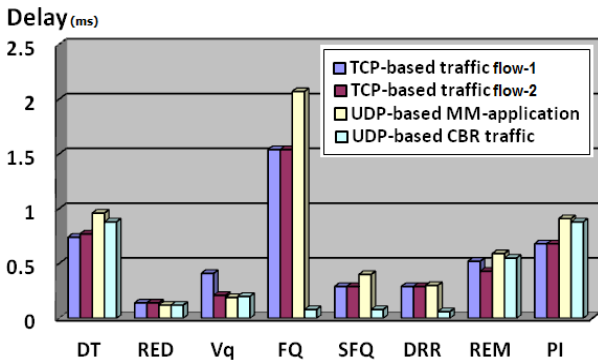


Figure 3. Delay of different traffic flows at different AQM algorithms

J. Packet Loss

The Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packets may be lost or dropped in a network when a queue in the network router overflows. The amount of packet loss during the steady state is an important property of a congestion control scheme because the larger the value of packet loss, the smaller throughput. In this evaluation, the amount of packet loss is measured by the loss rate parameter that is calculated as the ratio of the lost packets to the total transmitted packets.

Fig. 4 shows the packet loss rate of different traffic flows when applying different AQM schemes DT, RED, VQ, FQ, SFQ, DRR, REM and PI. From Fig. 4, the FQ, SFQ, and DRR drop the multimedia packets more frequently than other algorithms. Although these fair queuing algorithms provide fair buffer allocation between different traffic types, they do not guarantee the required QoS for real-time UDP-based multimedia applications.

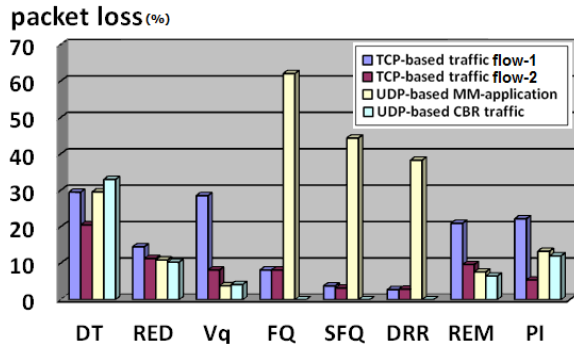


Figure 4. packet loss of different flows at different AQM algorithms

K. Fairness Index

Fairness index is the ratio of individual throughputs summation squared to the summation of individual throughputs squared multiplied by the number of traffic classes. Fig.5 shows the fairness index of different traffic flows achieved by applying different AQM schemes DT, RED, VQ, FQ, SFQ, DRR, REM and PI.

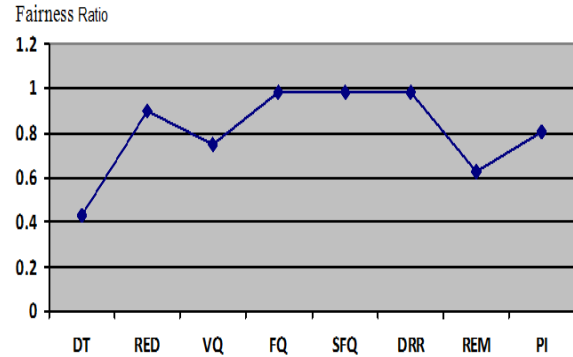


Figure 5. Fairness index of different AQM algorithms

Fig.5 indicates that the DT and REM algorithms provide the lowest fairness index while the FQ, SFQ, and DRR achieve the same fairness.

L. Global Power

AQM algorithms are used to provide high throughput with low delays. The ratio of the throughput to the delay is called global power [20]. It is desirable to provide high global power values as possible. Table I and as Fig. 6 show the global power achieved for different traffics by different AQM schemes. From the results, RED provides the highest power for the multimedia applications, while FQ gives the lowest global power for multimedia traffics.

TABLE I. GLOBAL POWER OF DIFFERENT TRAFFIC FLOWS AT DIFFERENT AQM ALGORITHMS

	TCP-based flow1	TCP-based flow2	UDP-based (multimedia)	UDP-based (CBR)
DT	7.96	5.6	66.52	13.57
RED	114.71	114	273	133.17
VQ	10.1	126	186.11	85.4
FQ	15.1	15.08	11.12	222.8
SFQ	82.34	80.69	55.55	222.8
DRR	81.34	79.62	76.13	297
REM	14.44	28.93	85.19	30.3
PI	15.29	11.75	58.52	17.82

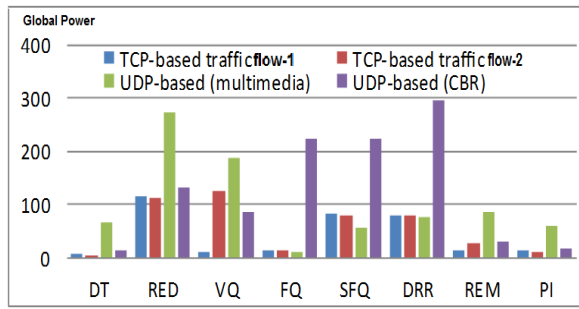


Figure 6. Global power of different traffic flows at different AQM algorithms

Table II summarizes the performance evaluation. In Table 2, Flow 1 and Flow 2 are TCP-based flows (FTP), Flow 3 is a UDP-based flow (multimedia traffic video/audio) and Flow 4 is a UDP-based traffic (CBR). The table shows the simulation results of throughput, delay, the percentage packet loss and fairness. From Table II, the DRR has the lowest loss rate for TCP-based flows, FQ has higher loss rate compared with other algorithms especially for multimedia traffics that is punished more than other flows, and this is the matter with SFQ or DRR. The REM has good performance in terms of packet loss and average queuing delay, as shown in Table II.

TABLE II. AVERAGE THROUGHPUT, DELAY AND PACKET LOSS OF DIFFERENT TRAFFIC FLOWS AT DIFFERENT AQM ALGORITHMS

		<i>DT</i>	<i>RED</i>	<i>VQ</i>	<i>FQ</i>	<i>SFQ</i>	<i>DRR</i>	<i>REM</i>	<i>PI</i>
<i>Average throughput (Kbytes/s)</i>	<i>Flow 1</i>	5.89	16.06	4.14	23.26	23.88	23.59	7.51	10.4
	<i>Flow 2</i>	4.33	19.74	26.44	23.23	23.4	23.09	12.44	7.99
	<i>Flow 3</i>	63.86	32.76	35.36	23.02	22.22	22.84	50.26	53.26
	<i>Flow 4</i>	11.94	15.98	17.08	17.82	17.82	17.82	16.66	15.68
<i>Average delay (ms)</i>	<i>Flow 1</i>	0.74	0.14	0.41	1.54	0.29	0.29	0.52	0.68
	<i>Flow 2</i>	0.77	0.14	0.21	1.54	0.29	0.29	0.43	0.68
	<i>Flow 3</i>	0.96	0.12	0.19	2.07	0.4	0.3	0.59	0.91
	<i>Flow 4</i>	0.88	0.12	0.2	0.08	0.08	0.06	0.55	0.88
<i>System throughput (kbps)</i>		924.2	676.43	664.15	698.8	699.09	699.22	695.38	699.09
<i>Loss rate (%)</i>	<i>Flow 1</i>	29.53	14.5	28.57	8.13	3.77	2.74	20.96	22.2
	<i>Flow 2</i>	20.53	11.3	8.16	8.13	3.18	2.88	9.65	5.41
	<i>Flow 3</i>	29.58	10.9	3.81	62.04	44.37	38.28	7.6	13.28
	<i>Flow 4</i>	33	10.3	4.15	0.0	0.0	0.0	6.5	12.01
<i>Total loss (%)</i>		28.21	0.11	6.5	30.72	18.78	15.62	8.8	11.62
<i>Fairness index</i>		0.43	0.9	0.75	0.986	0.988	0.989	0.63	0.81

It is clear from Table II that DRR offers equal throughput to all kinds of sources like other fair queuing algorithms (FQ & SFQ). In addition, it has the lowest loss rate as shown in Table II.

M. Discussion

It clear from the above results that the network performance is sensitive to the applied queue management algorithm. Each algorithm improves some QoS aspects while it degrades the quality of service delivered by other parameters. Drop Tail (DT) suffers from the full queue problem, unfair resource allocation, lockout behavior, global synchronization, low throughput, and long delays with high losses associated with different traffic types.

RED performs well with TCP traffic flow. Nevertheless, it suffers from poor fairness for mixed traffic network, as shown in Fig. 5 and Table II. RED reveals the critical problem that unresponsive UDP-based traffic flows, which may have greedier flow-control mechanisms than TCP-based flows, take more share of the output bandwidth. Non-TCP flows, especially for

unresponsive ones that may monopolize the output bandwidth. This is because TCP connections is respond to congestion by reducing their sending rates, and are exposed to the same drop rate. The bad effects of this problem may be harmful especially when the number of multimedia traffics increases. Therefore, when considering other types of traffic that differ from TCP-based traffic, RED may suffer from lack of fairness [9]. Other problems associated with RED are the average queue size dependent on the parameter settings, unpredictable average queuing delay, and the sensitive performance to the traffic load. VQ suffers from the lockout problem, and high delays compared to RED, as shown in Table II.

The primary benefit of FQ is that each flow has its own queue. Therefore, an extremely misbehaving flow does not degrade the quality of service delivered to other flows. However, if a flow attempts to consume more than its fair share of bandwidth, then only its queue is affected. Hence, there is no impact on the performance of the other queues on the shared output port. The FQ experiences more delays; it is more aggressive toward multimedia



traffic applications that are delay sensitive as indicated in Fig. 3. The drawbacks of FQ are the packet length independency; the packet is served in its turn regardless of its length. In addition, it is more complex than other strategies and it cannot handle different flows bandwidth requirements. SFQ improves delay performance with respect to FQ algorithm, but it exhibits more packet loss as in the DRR. The DRR has complexity $O(1)$. However, due to its round robin structure, its latency properties are not adequate for latency-critical applications, such as voice [17].

REM works well with Explicit Congestion Notification (ECN) [16]. It achieves high utilization with negligible loss or queuing delay even when the load increases. However, REM is more complex due to parameter settings. In addition, it provides low throughput for Web traffics.

Like REM, the PI performs well when used in conjunction with the ECN. However, it has some limitations and drawbacks due to the model error introduced by linearization. Comparing with REM, the PI performs better in terms of TCP-based traffic throughput and delay, but they maintain approximately the same performance for the multimedia traffic applications.

4. DRAWBACKS OF CURRENT SCHEMES

Although many algorithms were developed to solve the network congestion problem, they handle different types of traffics using the same strategy. Nevertheless, with the explosive growth of the internet real-time and non-real-time activities, it is a difficult emotion to guarantee required QoS. This is because different traffics need different QoS requirements. Thus, special measures, such as quality-of-service routing, must be taken to keep packets from being dropped. In other words, for congestion control and avoidance strategies, it is more robust to decouple the congestion measure with the traffic load to be able to respond effectively to different network conditions and to guarantee QoS requirements for different traffic classes. Approaches to the problem of fairness, such as FQ, SFQ, and DRR, employ per-flow queuing. Although they provide fair allocation of bandwidth and lower delays, they penalize UDP-based traffic applications that need specific QoS requirements. Therefore, we argue that AQM based congestion control should be adaptive to the dynamically changing traffic situation in order to detect, control and avoid current and incipient congestion proactively.

5. PROPOSED APPROACH

This section presents a new RED-based AQM algorithm called Dynamic Queue RED (DQRED). The aim is to guarantee the QoS requirements for real-time applications, and to prevent starvation of the best effort traffic (TCP-based applications) in the presence of non-TCP-based applications (real-time traffics) especially at heavy traffic loads.

The main idea of the proposed DQRED strategy is similar to that developed in [21] with additional modifications allowing packets from different traffic classes to be scheduled (served) dynamically according to the number of active flow packets (enqueued) in each queue. In [21], a static priority scheduling approach is developed by using three queues buffer, one queue for each traffic class type. Where, the incoming packets are classified into three classes; UDP-based video traffic with high priority, UDP-based audio traffic with medium priority, and TCP-based traffic with low priority. These different traffic classes are serviced by static priority scheduling using ratio 3:2:1, That is the scheduler transmits three video packets, then two packets of the audio, and finally after serving video and audio traffic classes one data packet. In this paper, the scheduler services different packets dynamically. Figures 7, 8, 9 and 10 illustrate the idea and behavior of the proposed DQRED.

As shown in Fig. 7, the proposed algorithm employs three queues in the router and classifies the incoming traffic packets into three classes according to their class types, as that developed in [21]. However, in the proposed DQRED, the queued packets are scheduled dynamically based on the number of active packets already existing in the queues during a predefined time. The proposed DQRED isolates multimedia video or audio traffic applications (real-time) from best-effort traffic by using a specific header bit. Fig. 8 shows the operation of the classifier that performs the classification process. In addition, the proposed DQRED dynamically schedules packets of each class type based on the number of packets that already exist in the queues during a predefined time interval. This count is updated periodically. The flowchart in Fig. 9 shows how the scheduler operates and how the packets are served in the DQRED. Using multiple queues in the router (one for each individual class type) enables the scheduler to provide the QoS requirements for real-time traffics. The implementation code of the proposal DQRED is shown in Fig. 10.

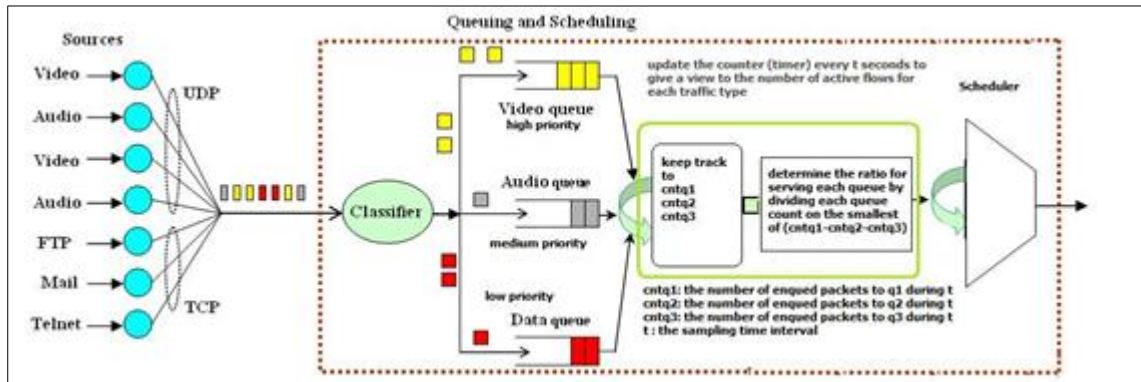


Figure 7. Queuing and Scheduling strategies of DQRED

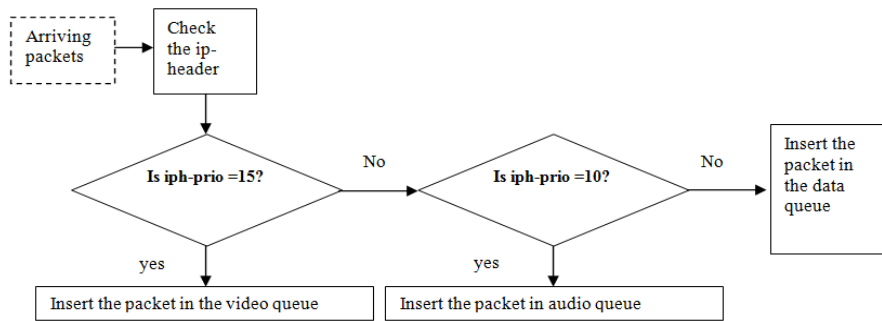


Figure 8. Classifier operation of the proposed DQRED

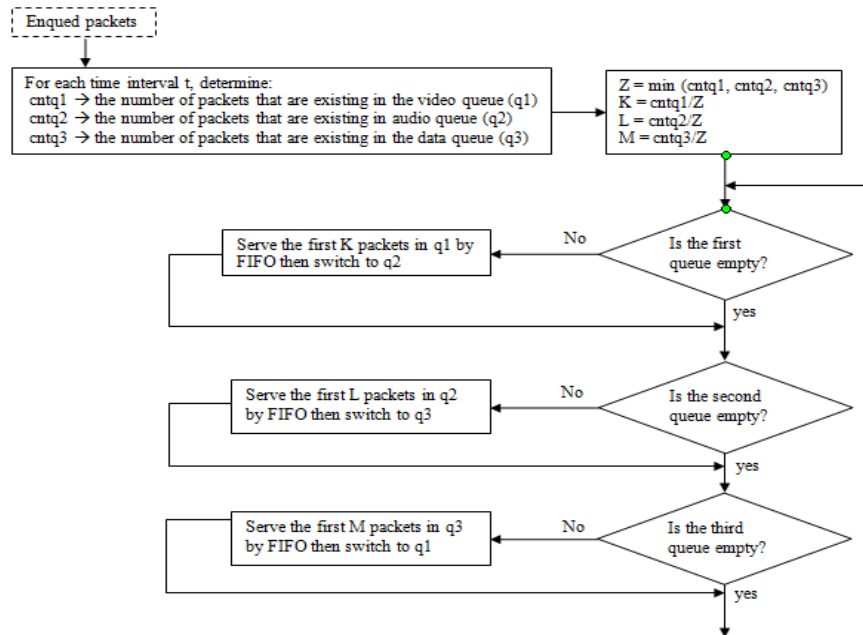


Figure 9. Scheduler operation of the proposed DQRED

Figure 10.

6. PERFORMANCE EVALUATION OF THE PROPOSED DQRED APPROACH

This section presents a performance evaluation of the proposed algorithm by using the Network simulator NS-2 [10, 11]. It also presents a comparative study between the proposed DQRED algorithm and the Random Early Detection (RED) algorithm. In this evaluation, the AT&T realistic topology, shown in Fig. 11, is used to test the performance of the proposed strategy. The AT&T network topology is created by using the network topology generator GT-ITM [22, 23]. This AT&T topology contains 166 nodes and 189 links. In this study, the performance of the proposed DQRED algorithm is evaluated by measuring various QoS metrics at different traffic loads, i.e., different numbers of input sources N (60, 70, 100, and 150). Table III shows the number of sources belonging to each traffic type (data, audio, and video) at each value of N . The simulation time is 40 seconds.

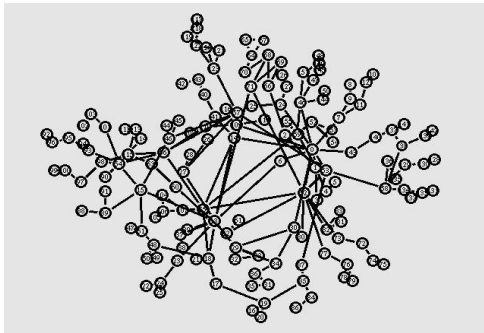


Figure 11. T & T network topology

TABLE III. NUMBER OF DATA, AUDIO, AND VIDEO TRAFFIC SOURCES AT DIFFERENT N VALUES

<i>Number of sources N</i>	60	70	100	150
<i>TRAFFIC TYPE</i>				
FTP data application	20	30	30	30
Audio traffic application	20	20	30	50
Video traffic application	20	20	40	70

N. Effects of using DQRED on throughput

Fig. 12 shows the network throughput for the FTP data applications when using the proposal DQRED and RED algorithms. From Fig. 12, the proposed DQRED improves the performance of data throughput especially at heavy traffic loads ($N=150$).

Initialization:

```
Define three queues instead of one queue buffer
avg1, avg2, avg3 = 0
count1, count2, count3 = 1
cntq1, cntq2, cntq3 = 0
```

For each packet arrival (the enqueue event):

```
(1) Check the ip-header:
    if iph -> prio_ = 15 → enqueue the packet in q1 ( video
    application)
    if iph -> prio_ = 10 → enqueue the packet in q2 (Audio
    application)
    else →enqueue the packet in q3 (TCP-based traffic
    applications)
(2) Calculate the new average queue size for all queues ('avg1',
'avg2', 'avg3')
    if the queues is non-empty
        avg_i = (1-wq) * avg_i + wq * q_i // i refers to the queue for
        each class
    else
        m = f(time - q_i_time)
        avg_i = (1-wq)^m * avg_i
        if minth <= avg_i < 2*maxth
            increment count_i
            increment cntq_i
        if avg_i < maxth (calculate probability 'pb') // as in
        normal RED
            pb = maxp * (avg_i - minth) / (maxth - minth)
        else if maxth <= avg_i < 2 * maxth
            pb = (avg_i - maxth) * (1-maxp) / maxth + maxp // as
        in GRED

        with the probability 'pa' ==> mark the arriving packet
        pa = pb / (1 - count_i * pb)
        count_i = 0     else     count_i = -1
    when queue becomes empty
        q_i_time = time
```

For each time interval 't' (the deque event):

```
1- determine the values of cntq1, cntq2, cntq3 through the
first second
2- define z as z = the smallest of (cntq1, cntq2, cntq3)
3- calculate the number of packets that will be served from
each queue as:
    K = cntq1/z & L = cntq2/z & M = cntq3/z
4- the scheduler will serve the enqueued packets as follow:
    After serving K packets from q1, L packets from q2
    are served, then finally M packets from q3 are being
    served
5- after t seconds the values for K, L, and Mare updated
and so on.
```

Figure 12. Mplementation the proposed DQRED

Fig. 13 and Fig. 14 show the network throughput for the audio and video traffics when using the DQRED and RED algorithms. From the figures, the proposal DQRED improves the throughput of both audio and video traffics. This is because; the proposed DQRED algorithm serves different traffics according to the queued packets of each type.

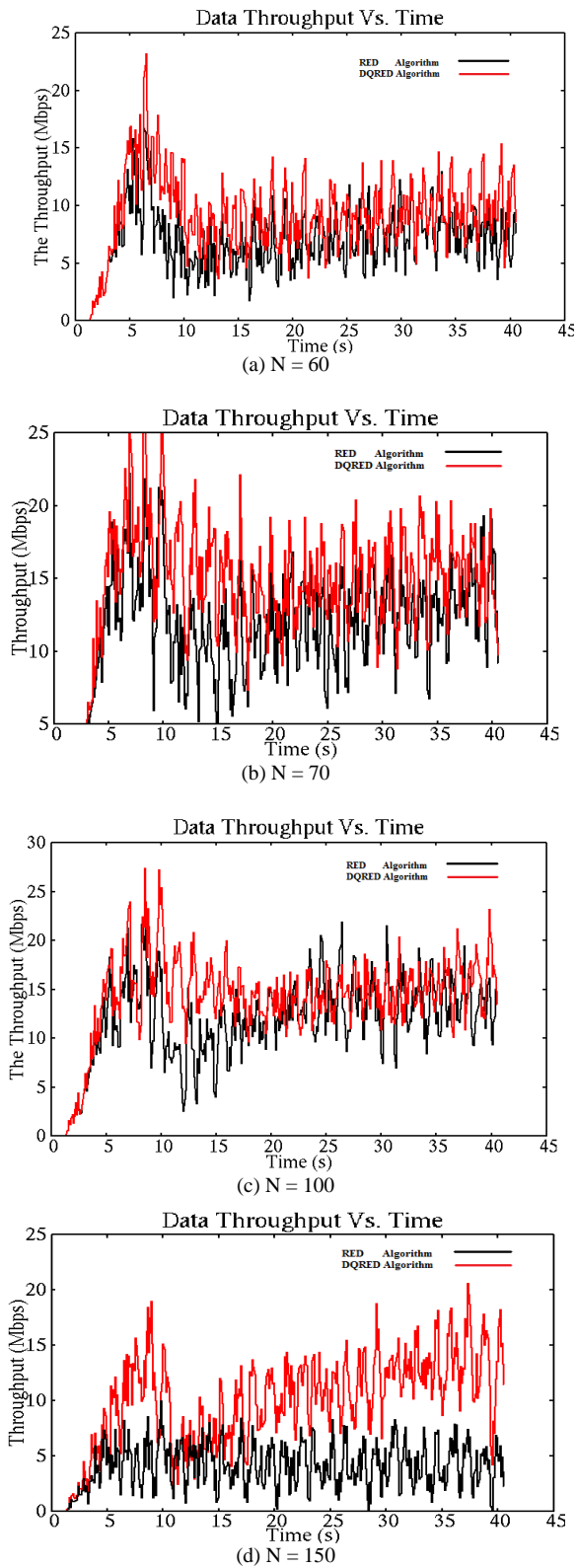


Figure 13. Ata throughput when using

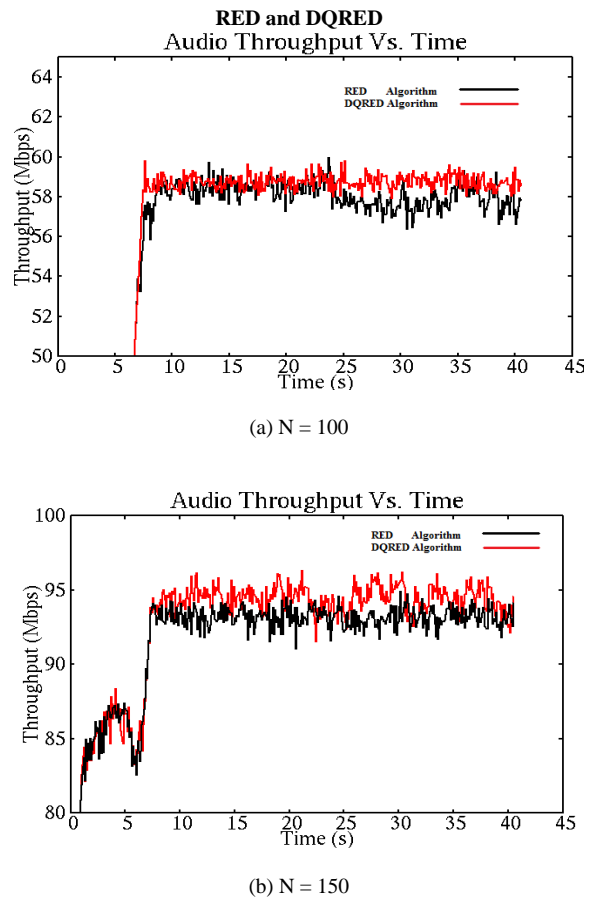


Figure 14. Udio throughput when using RED and DQRED Different values of N

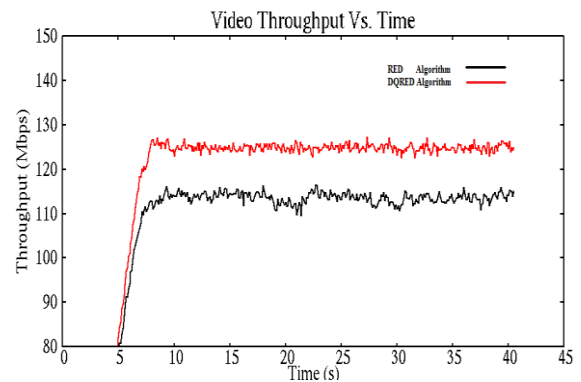


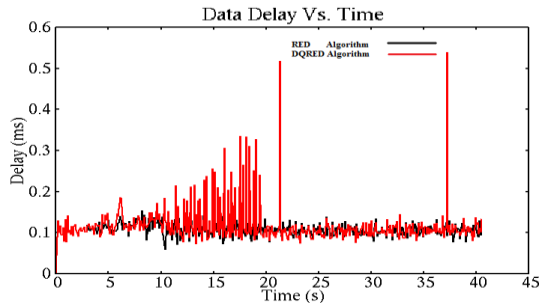
Figure 15. Ideo throughput when using RED AND DQRED algorithms at $N=150$

O. Effects of using DQRED on delay

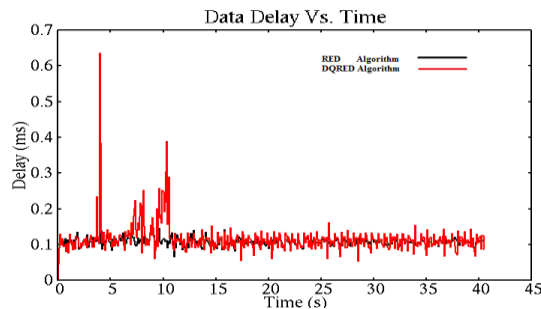
Fig. 15 shows the data packet delay when using the DQRED and RED algorithms. The average delay elapsed by both the DQRED and RED algorithms is similar at high traffic loads ($N = 150$), as shown in Fig. 15(b). While



evinced, shows the data delay elapsed by the DQRED algorithm is slightly greater than that obtained when using the RED algorithm, especially at low traffic loads ($N = 60$), as shown in Fig. 15(a).



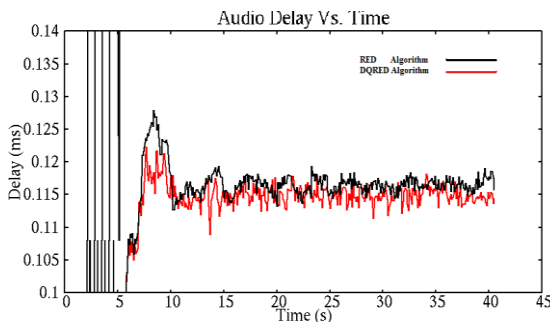
(a) $N = 60$



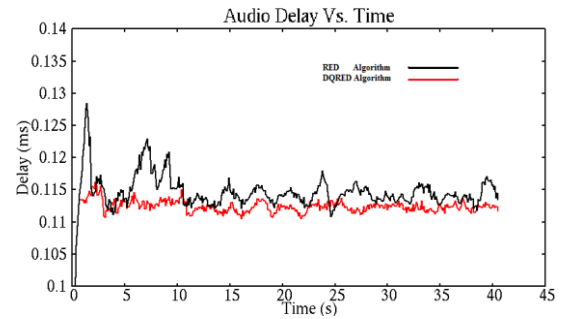
(b) $N = 150$

Figure 16. ata packets delay when using RED and DQRED algorithms

Fig. 16 and Fig. 17 show the packet delay for the audio and video traffics respectively when using the DQRED and RED algorithms. The results indicate that as the number of input sources increased, the audio and video delay increased. In addition, the results indicate that the proposal DQRED gives good delay for both the video and audio traffic applications at heavy traffic loads.

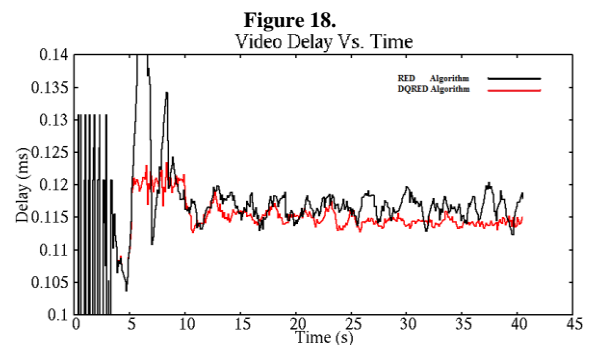


(a) $N = 60$

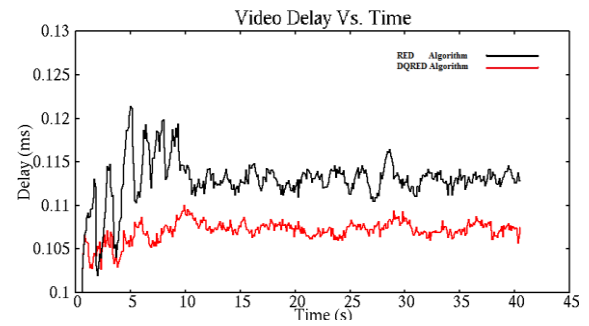


(b) $N = 150$

Figure 17. Audio delay when using RED and EQRED algorithms



(a) $N = 60$

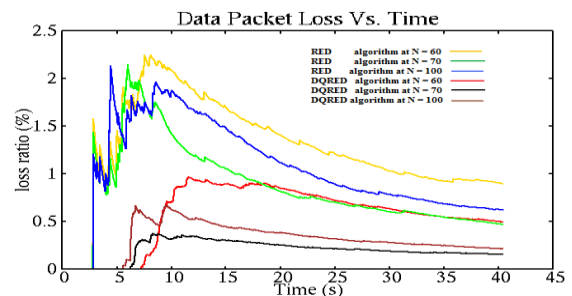


(b) $N = 150$

Figure 19. Video delay when using RED and DQRED algorithms

P. Effects of using DQRED on packet loss

Fig. 18 shows the data packet loss ratio when using the DQRED and RED algorithms. As shown in Fig. 18 (a), with RED algorithm, the data packet loss ratio is higher than that of DQRED algorithm at different loads.



(a) Packet loss ratio at different values of N

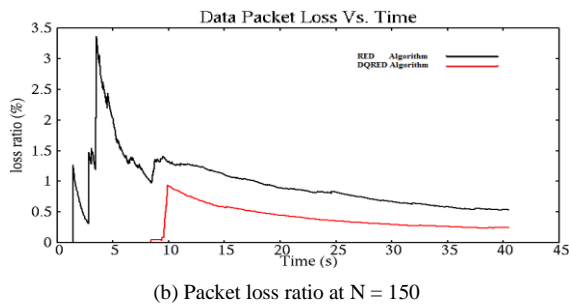
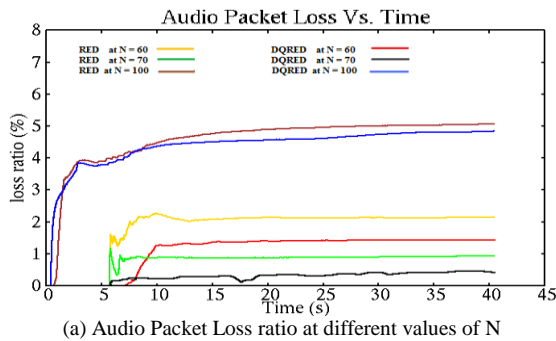
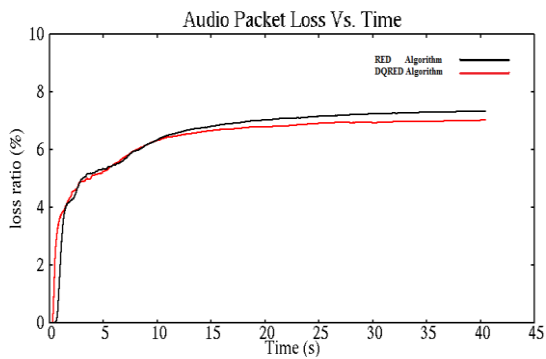


Figure 20. Data packet loss ratio when using RED and DQRED

In addition, when the number of sources increased ($N = 150$), the DQRED approach provides low data loss ratio, as shown in Fig. 18 (b). Fig. 19 shows the packet loss ratio for audio traffic when using both the DQRED and RED algorithms. As shown in Fig. 19 (a), the DQRED algorithm provides lower loss ratio than the RED algorithm at different values of traffic sources N . At $N=150$, the DQRED algorithm provides lower loss ratio than the RED algorithm as shown in Fig. 19 (b).



(a) Audio Packet Loss ratio at different values of N

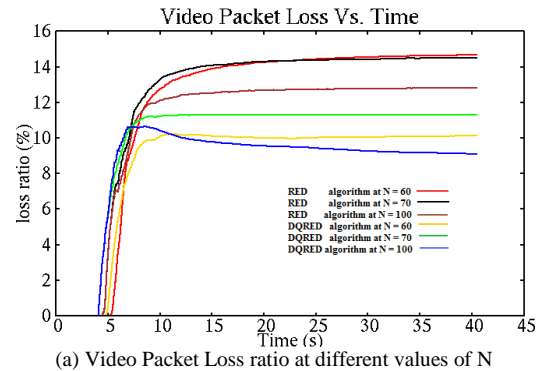


(b) Audio Packet Loss ratio at $N = 150$

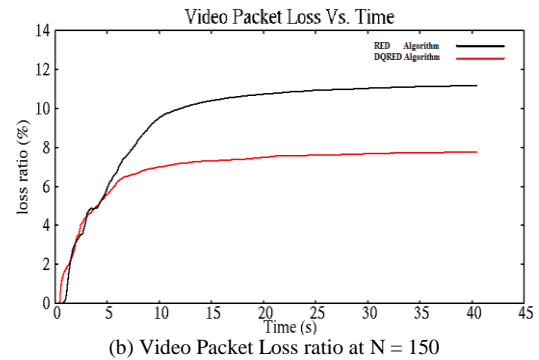
Figure 21. audio packet loss ratio when using RED and DQRED algorithms

Fig. 20 shows the packet loss ratio for video traffic when using both the DQRED and the RED algorithms. As shown in Fig. 20 (a), the DQRED algorithm provides lower loss ratio than the RED algorithm at different values of traffic sources N . In addition, at heavy traffic load

$N=150$, the DQRED algorithm provides lower loss ratio than the RED algorithm as shown in Fig. 20 (b).



(a) Video Packet Loss ratio at different values of N



(b) Video Packet Loss ratio at $N = 150$

Figure 22. Video packet loss when using RED and DQRED algorithms

7. CONCLUSION AND FUTURE WORK

In this paper, a new RED-based AQM algorithm is developed to guarantee the QoS requirements for the multimedia real-time applications. In addition, it prevents starvation of the best effort traffic (TCP-based applications) in the presence of non-TCP-based traffics (real-time applications) especially at heavy traffic loads. The proposed algorithm first classifies the incoming packets into three class types, each of which is handled by one of three queues in the internet router. The queued packets are then scheduled dynamically based on the number of active packets already existing in the queues during a predefined time interval. The simulation results show that dynamic scheduling of queued packets improves the performance of multimedia traffic application in aspects of throughput, delay, and packet loss, especially at heavy loads without affecting the performance of the TCP-based traffic applications.

In the future work, other parameters such as incoming traffic rate and available network bandwidth will be considered in the dynamic weight adjustment of the



packet scheduler of the proposed DQRED to improve the QoS of the real-time traffic applications.

ACKNOWLEDGMENT

The authors extend their thanks to Dr. Heba El-Khobby (Faculty of Engineering – Tanta University) for the comments and suggestions that helped in improving the quality of the proposed strategy.

REFERENCES

- [1] N. Hamadneh, D. Murray, M. Dixon, and P. Cole, "Dynamic Weight Parameter for the Random Early Detection (RED) in TCP Networks", *International Journal on New Computer Architectures and Their Applications (IJNCAA)*, Vol. 2, No. 2, pp. 342-352, 2012.
- [2] V. Jacobson, and M. J. Karels, "Congestion Avoidance and Control," *Proceedings of the ACM SIGCOMM*, Vol.18, No. 4, pp. 314-329, August 1988.
- [3] J. Y. BOUDEC, "Rate adaptation, Congestion Control and Fairness: A Tutorial," *Ecole Polytechnique F'ed'erale de Lausanne (EPFL)*, December 9, 2008.
- [4] R. Adams, "Active Queue Management: A Survey", *IEEE Communication Surveys & Tutorials*, Vol.15, No. 3, 2013.
- [5] G. Patil, S. McClean and G. Raina, "Drop Tail and RED Queue Management with Small Buffers: Stability and HOPF bifurcation", *ICTACT Journal on communication technology: special issue on next generation wireless networks and applications*, Vol. 2, Issue 2, June 2011.
- [6] G. Thiruchelvi and J. Raja, "A Survey on Active Queue Management Mechanisms", *International Journal of Computer Science and Network Security*, Vol. 8, No.12, December 2008.
- [7] K.Chitra, and G.Padamavathi, "Classification and Performance of AQM-Based Schemes for Congestion Avoidance", *International Journal of Computer Science and Information Security*, Vol. 8, No. 1, April 2010.
- [8] M. Cheng, and X. Ma, "Performance Evaluation of Queue Management Methods for Congestion Control", *Journal of Information & Computational Science*, Vol. 9, No. 6, pp.1599–1608, 2012.
- [9] S. Kochher,G. Singh, "Comparison of Various Queuing Techniques to Select Best Solution for Eradicating Congestion in Networks", *International Journal of Engineering and Computer Science*, Vol. 03, Issue 07, pp. 7010-7017, July 2014. ISSN: 2319-7242
- [10] G. Wang, Y. Xia, and D. Harrison, "An NS2 TCP Evaluation Tool: Installation Guide and Tutorial" V0.2, November 24, 2008.
- [11] S. McCanne and S. Floyd, "ns Network Simulator", <http://www.isi.edu/nsnam/ns>.
- [12] M. Abu Obaida, Md. Sanallah Miah, Md. Abu Horaira, "Random Early Discard (RED-AQM) Performance Analysis in Terms of TCP Variants and Network Parameters: Instability in High-Bandwidth-Delay Network", *International Journal of Computer Applications* (0975 - 8887), Vol. 27, No. 8, pp. 40-44, August 2011.
- [13] S. Kunniyur, and R. Srikant, "An Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management," *IEEE/ACM Transactions on Networking*, Vol. 12, No. 2, pp. 286-299, 2004.
- [14] H. Wang, C. Liao, and Z. Tian, "Effective Adaptive Virtual Queue: A Stabilising Active Queue Management Algorithm for Improving Responsiveness and Robustness", *IET Communications*, Vol. 5, No. 1, pp.99-109, 2011.
- [15] A. Ghodsi, V. Sekar, M. Zaharia, I. Stoica, "Multi-Resource Fair Queueing for Packet Processing", *SIGCOMM'12*, August 13–17, 2012, Helsinki, Finland.
- [16] D. Kalav, and S. Gupta, "Congestion Control in Communication Network Using RED, SFQ and REM Algorithm", *International Refereed Journal of Engineering and Science*, Vol. 1, Issue 2, pp.41-45, October 2012. ISSN (Online) 2319-183X, (Print) 2319-1821.
- [17] Luciano Lenzini, Enzo Mingozzi and Giovanni Stea, "Performance Analysis of Modified Deficit Round Robin schedulers," *Journal of High Speed Networks* Vol. 16, Issue 4, pp 399-422, October 2007.
- [18] S. Chebli, A. Elakkary, N. Sefiani and N.Elalami, "New PI stabilization of first-order congestion control of active queue management routers," *International Conference on Electrical and Information Technologies (ICEIT)*, 25-27 March 2015. DOI: [10.1109/EITech.2015.7162961](https://doi.org/10.1109/EITech.2015.7162961)
- [19] S.Chebli, A.Elakkary, N.Sefiani, and N.Elalami "PI Stabilization for Congestion Control of AQM Routers with Tuning Parameter Optimization," *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 4, No.1, pp 52-55, 2016. DOI: 10.9781/ijimai.2016.4110
- [20] W. Jiang, "An adaptive active queue management algorithm in internet," *MSc Thesis in computer Engineering*, Quebec University, 2006.
- [21] G. Attiya and H. El-Khobby "Improving Internet Quality of Service through Active Queue Management in Routers," *International Journal of Computer Science Issues*, Vol. 9, Issue 1, No 2, pp.279-286, January 2012.
- [22] O. Heckmann, M. Piring, J. Schmitt, and R. Steinmetz, "On Realistic Network Topologies for Simulation", August 2003.
- [23] O. Heckmann, M. Piring, J. Schmitt, and R. Steinmetz, "How to use Topology Generators to create realistic Topologies", *KOM Technical Report 07/2002*.



Sami A. El-dolil, received B. Sc. and M. Sc. degrees in electronic engineering from Menoufia University, Menouf, Egypt, in 1977 and 1981, respectively. In 1986 he joined the Communication Research Group at Southampton University, Southampton, England, as a research student doing research on teletraffic analysis for mobile radio communication. He received his Ph. D degree from Menoufia University, Menouf, Egypt, in 1989. He was a Post Doctor Research Fellow at the Department of Electronics and Computer Science, University of Southampton, UK, 1991-1993. From 1994 to 2008 he worked as an Associate Professor, and since 2008 up to now he is working as a professor at the Department of Electronics and Electrical Communication, Faculty of Electronic Engineering, Menoufia University. His current research interests are in high-capacity digital mobile System, and multimedia networks, Telecommunication systems network spectrum management, planning and optimization.



Gamal M. ATTIYA graduated in 1993 and obtained his M.Sc. degree in computer science and engineering from Menoufia University, Egypt, in 1999. He received a PhD degree in computer engineering from University of Marne-La-Vallée, Paris-France, in 2004. He is currently Associate Professor at Computer Science and Engineering department, Faculty of Electronic Engineering, Menoufia University, Egypt. His main research interests include distributed computing, task allocation and scheduling, cloud computing, Big Data analysis, computer networks and protocols, congestion control, QoS, and multimedia networking.



Hanaa Mohammed graduated in 2000, from the Department of Electronics and Electrical Communications Engineering, Tanta University, Egypt, and she prepared to receive her M.Sc degree this year in computer science and engineering from Faculty of Electronic Engineering, Menoufiya University, Egypt. Her research interests are in the field of computer networks including congestion control, routing, mechanisms for resource management and QoS.