New Class-based Dynamic Scheduling Strategy for Self-Management of Packets at the Internet Routers

Hanaa Mohammed[1], Gamal Attiya[2], and Samy El-Dolil[3]
[1]Department Electronics and Electrical Communications Engineering, Tanta University, Egypt
[2]Department Computer Science and Engineering, Menoufia University, Egypt
[3]Department Electronics and Electrical Communications Engineering, Menoufia University, Egypt

Abstract: *Recently, the Internet became the most important environment for many activities including sending emails, browsing web sites, making phone calls and even having a videoconference for far education. The incremental growth of the internet traffic leads to a serious problem called congestion. Several Active Queue Management (AQM) algorithms have been implemented at the internet routers to avoid congestion before happening and solve the congestion if it happens by actively controlling the average queue length in the routers. However, most of the developed algorithms handle all the traffics by the same strategy although the internet traffics, real time and non-real time; require different Quality of Service (QoS). This paper presents a new RED-based algorithm, called Dynamic Queue RED (DQRED), to guarantee the required QoS of different traffics. In the proposed algorithm, three queues are used in the internet router; one queue for each traffic type (data, audio and video). The arrived packets are first queued in the corresponding queue. The queued packets are then scheduled dynamically according to the load (the number of queued packets) of each class type. This strategy guarantees QoS for real time applications as well as service fairness.*

Keywords: *Congestion control, AQM, packet queuing, dynamic scheduling, multimedia QoS.*

Received December 31, 2015; accepted July 4, 2016

1. Introduction

Over the last few years, the Internet has experienced an explosive growth and became the most important environment for many activities including real time and non-real time applications. The incremental growth of the internet traffic leads to a serious problem called congestion [14]. This problem leads to performance degradation particularly for real-time applications; long delays in data delivery, more lost or dropped packets, and degradation of link utilization. Therefore, one of the keys to the success of the Internet is rely on using congestion control mechanisms. The main objective of such mechanisms is to keep the network running pretty close to its rated capacity, even when faced with extreme overload.

Various congestion control approaches have been proposed, either at the end hosts or at the routers, to solve the congestion problem. The approaches at routers use the router queue size to monitor the congestion state of the network. Two classes of router algorithms, related to congestion control, are presented; queue management and scheduling [20]. Queue management algorithms control the length of packet queues by dropping (or marking) packets when necessary, while scheduling algorithms are used primarily to determine how the bandwidth is allocated among flows by determining which packet to send next.

Queue management refers to the control that takes place in network routers. It may be categorized into Passive Queue Management (PQM) and Active Queue Management (AQM). The PQM does not take any preventive packet drop before the router buffer gets full or reaches a specified value while the AQM employs preventive packet drop before the router buffer gets full. The AQM mechanisms are used to notify the end systems by the required congestion information early enough to reduce their transmission rate before the buffer overflows. They have been proposed to prevent router buffer overflow and to avoid congestion. Random Early Detection (RED) is the most popular strategy of the AQM [19]. The RED uses a weighted-average queue size as a measure of congestion. When this size is smaller than a minimum threshold, no packets are marked or dropped. When the average queue length is between the minimum threshold and the maximum threshold, the probability of marking or dropping packets varies linearly between zero and a maximum drop probability. If the average queue length exceeds a maximum threshold, all packets are dropped to reduce the congestion.

Although the RED has been deployed in many routers, it faces numerous problems. There are no fixed set of rules to tune its parameters for a given network environment. Therefore, many RED-based strategies are presented in the literatures to rectify these problems and several types of AQM techniques are developed. These techniques include; Adaptive RED [9], Stabilized RED [18], Nonlinear RED (NLRED) [24], Gentle RED [2], Dynamic RED [8], Adaptive Virtual Queue (AVQ) [23], Enhanced RED (ENRED) [13], BLUE [8] and Balanced RED [3]. Fair Queuing (FQ) [11, 22], Stochastic Fair Queuing (SFQ) [17], Class-Based Scheduling (CBQ) [10], Random Exponential Marking (REM) [5], and Deficit Round Robin (DRR) [21] that ensure fair access to network resources and prevent a bursty flow from consuming more than its fair share. For more details, the reader may refer to [6].

Although numerous AQM techniques have been proposed, the most well-known AQM is RED [19]. This approach, while necessary and powerful with the Transmission Control Protocol (TCP) flows, is not sufficient to provide good service in all circumstances especially with non-TCP flows such as multimedia traffic. RED with its variants may suffer from lack of fairness when considering other types of traffic that differ from TCP-based traffic. Approaches to the problem of fairness, such as Fair Random Early Drop (FRED) [15] and Enhancement of Fair Random Early Detection [1], punish misbehaved non-TCP flows. These punishment mechanisms result in a poor performance for multimedia flows that are well behaved. Dynamic Class-Based Threshold (D-CBT) [7] improves multimedia performance on the Internet, where it overcomes the FRED shortcoming of punishing non-TCP flows more than others do. However, the D-CBT complexity bounded its use as a suitable algorithm to improve the QoS of multimedia especially at high traffic loads.

This paper presents a new RED-based dynamic scheduling strategy to self-management of packets in the Internet routers. The proposed strategy is structurally similar to the congestion control mechanism presented in [4] with additional modifications to guarantee QoS requirements of different traffics and guarantee fairness of service. The proposed strategy is called Dynamic Queue RED (DQRED). In the proposed algorithm, three queues are used in the internet router; one queue for each traffic type (data, audio and video). The arrived packets are first classified and queued in the corresponding queue. The queued packets are then scheduled dynamically according to the number of queued packets of each class type. By isolating different packets at the router based on their class type and periodically changing the service ratio, the strategy guarantees QoS for real time applications as well as guarantees fairness of service. The proposed algorithm is evaluated and compared with the most recent algorithms by using the Network simulator NS-2. The simulation results show that dynamic scheduling enhances the performance of multimedia traffic application in aspects of throughput, delay, and packet loss without affecting on the performance of the TCP-based traffic application.

This paper is organized as follows. Section 2 introduces the problem of current AQM schemes. Section 3 presents a new queuing and scheduling strategy and describes the proposal DQRED strategy. Section 4 presents the simulation results. Finally, section 5 presents the concluding remarks.

2. Problem Definition

To date, AQM appears promising since the predominant transport protocol on the Internet is TCP. Unfortunately, the current queuing and scheduling strategies at the internet routers does not guarantee Quality of Service (QoS) for real time traffics. Briefly, the traffic characteristics of real-time and non-real-time applications require a certain QoS from the network in terms of bandwidth and delay requirements. Non-real-time applications are very sensitive to lost packets while real-time multimedia applications can tolerate some data loss, but are very sensitive to variance in packet delivery, called jitter. However, the current AQM strategies at the internet routers handle different packets of different traffics by the same strategy. In addition, most non-TCP flows (real time traffics) get an unfair share of network bandwidth when there is congestion. This unfairness occurs because many non-TCP flows do not reduce transmission rates when congestion occurs while the TCP flows are forced to transmit data at their minimum rates.

In summary, although numerous AQM techniques have been proposed, the major shortcoming of the current AQM strategies is that queue management itself does not provide per-flow state, which is required for achieving fairness among flows and guarantee QoS for real time traffics. This paper overcomes this problem by scheduling the queued packets at the Internet router dynamically based on the number of waited packets of each class type.

3. Proposed Strategy

The proposed strategy is structurally similar to the congestion control mechanism presented in [4] with additional modifications to guarantee QoS requirements of different traffics and guarantee fairness of service. In [4], the mechanism first classifies the input traffics into three classes; User Datagram Protocol (UDP-based) video traffic with high priority, UDP-based audio traffic with medium priority, and TCP-based data traffic with low priority. The mechanism then serves these different traffic classes by static priority scheduling ratio 3:2:1. This means that, the scheduler transmits (dequeue) three video packets, then two audio packets, and finally one data packet. In this paper, the input packets are first classified into three classes and queued into three queues as done in [4], but the queued packets are scheduled dynamically. That is, packets from different traffic classes are scheduled (served) dynamically according to the number of waited packets in each queue. The proposed strategy is called Dynamic Queue RED (DQRED). Figure 1 illustrates the main idea of the proposed DQRED. The DQRED uses three queues each of which handles an individual class of traffic whether real-time video, real-time audio or best-effort data traffic. The arriving packets are classified by the classifier based on a specific header bit, and each individual class packet is tracked into one of the three queues to be served or waited based on the available service. Then, the queued packets in different queues are scheduled dynamically based on the total number of queued packets in the queues. The scheduler calculates a service ratio periodically based on the number of waiting packets at different queues. Hence, each class will be served according to the calculated ratio at that period.

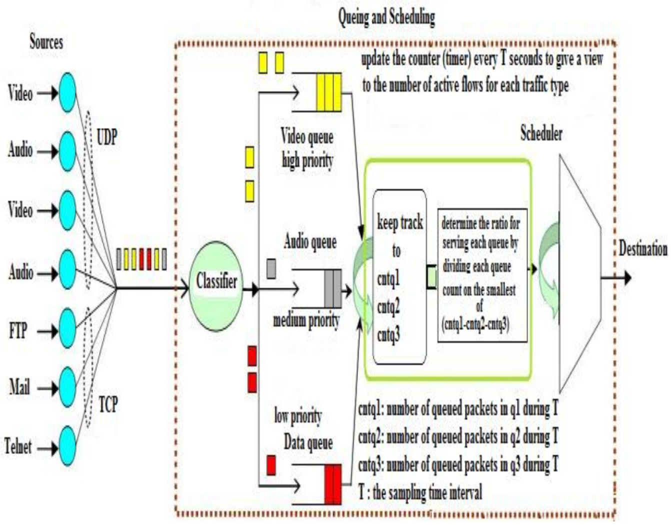


Figure 1. Queuing and scheduling strategy of the proposed DQRED.

DQRED achieves the following:

- Isolating real-time multimedia video or audio traffic from best-effort traffic, using multiple queues in the router one for each individual class type, enables the scheduler to provide the suitable service requirements for real-time traffic.
- Dynamic scheduling of packets belong to each class type based on the number of packets already exists in the queues periodically, guarantees fairness between different traffics.

4. Performance Evaluation

To evaluate the performance of the proposed DQRED strategy, it is coded in C++ and incorporated into the Network Simulator NS-2 [16] to be used as a congestion control mechanism in the routers instead of the normal RED. The performance of the proposed mechanism is evaluated by comparing the results obtained by the proposal DQRED with that obtained by the normal RED and other algorithms. The evaluation is done considering two simulation scenarios; a simple network topology and the real AT&T network topology.

4.1. Evaluation Metrics

4.1.1. Throughput

Throughput is the most widely used performance measure. In communication networks, throughput or network throughput is determined as the number of packets received successfully in 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. The throughput of different traffic flows can be used as an indicator to the fairness between these flows.

4.1.2. Delay and 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 very important factor of any network because:

1. Multimedia applications (audio and video applications) do not perform well if the delay is above a threshold value.
2. It is difficult to support many real-time applications in very large variation in delay (jitter).
3. 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.

4.1.3. Packet Loss

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packets can be lost in a network because they may be dropped 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 it is related to the throughput; the larger the value of packet loss, the smaller throughput is. In this paper, 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.

4.2. First Scenario: Simple Network Topology

Figure 2 shows a simple network topology to be used in this simulation study. In this topology, the queue limit of RED buffer is 10 packets; the min_{th} and the max_{th} are set to 5 and 15 respectively. The network performance is examined by considering the normal RED and the proposal DQRED at different values of input sources (N).

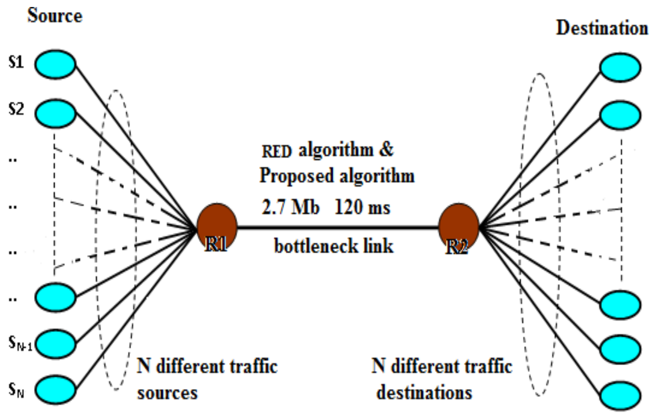
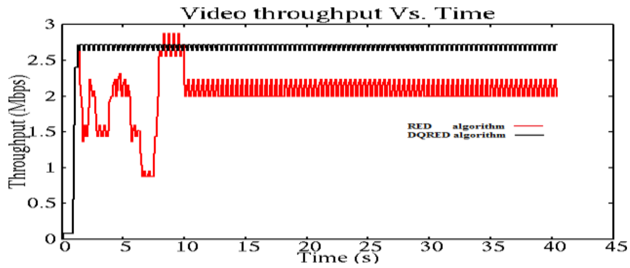


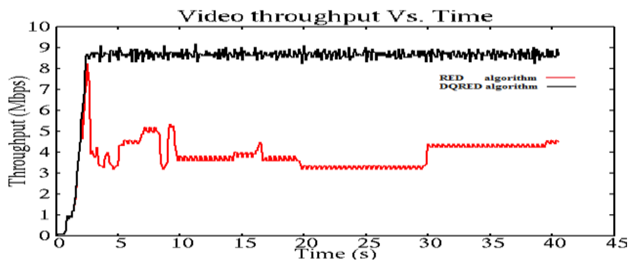
Figure 2. Simple network topology [23].

4.2.1. Effects of using DQRED on Throughput

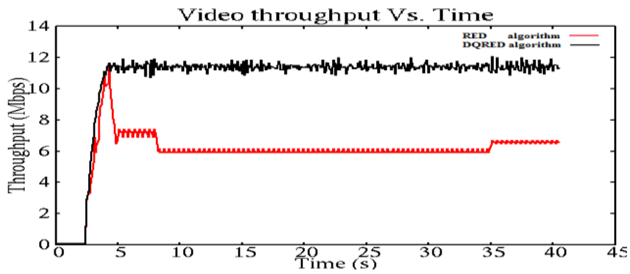
Figure 3 shows the instantaneous throughput of video traffic considering the RED and the DQRED at different values of N. From the figure, the proposed DQRED improves the throughput performance of the video traffic compared with that obtained by the RED.



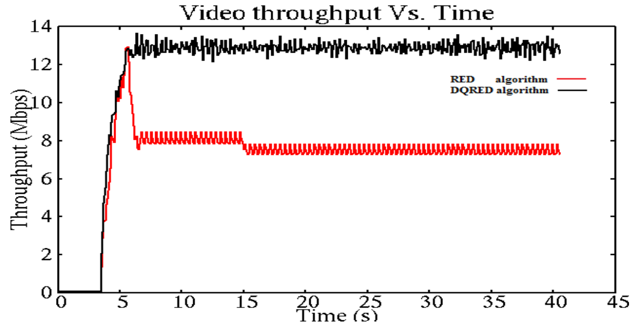
a) N = 10.



b) N = 30.



c) N = 60.



d) N = 75.

Figure 3. Throughput of video traffic at different values of N.

Figure 4 shows the average throughput of audio traffic considering the RED and the proposed DQRED at different values of N. The average throughput of audio traffic is slightly reduced with DQRED comparing with the normal RED. This behaviour is expected because the proposed DQRED tries to ensure fair bandwidth allocation between all traffic classes to provide QoS requirements of video traffic.

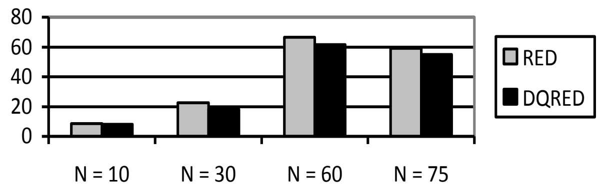


Figure 4. Average throughput (Mbps) of audio.

Figure 5 shows the average throughput of data traffic considering the RED and the DQRED at different values of N. From Figure 5, the RED outperforms the DQRED at small values of N. However, for large N, the DQRED performs as the same way of RED for data traffic applications. This means that, for large N, DQRED guarantees the performance of data traffic.

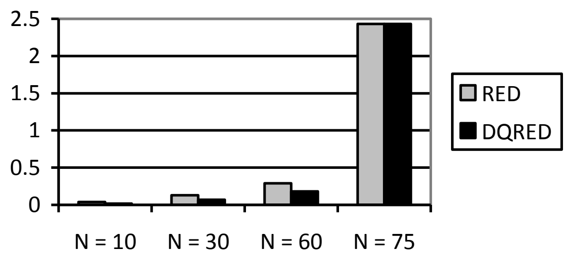
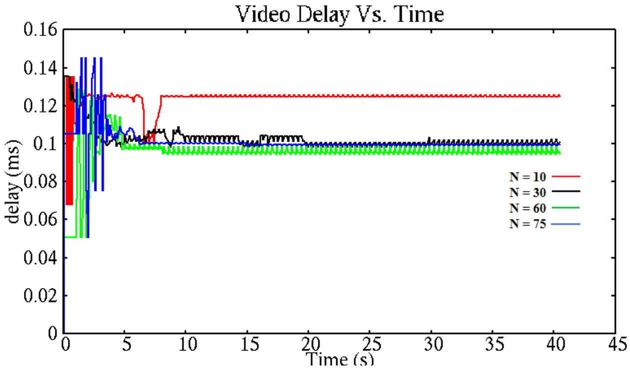


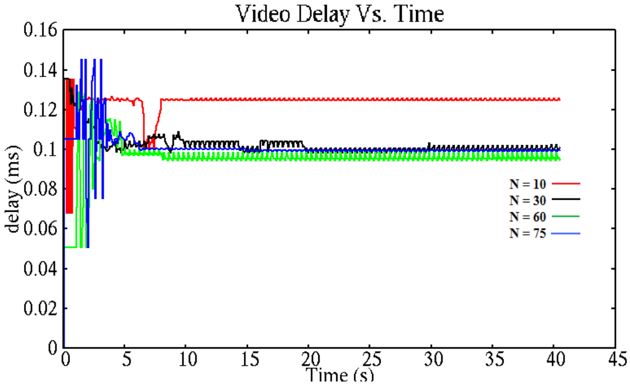
Figure 5. Average data throughput at different values of N.

4.2.2. Effects of Using DQRED on Delay

Figure 6 shows the instantaneous packet delay for video traffic considering at different values of N. The figures show that the DQRED provide a slightly stable delay at different values of N.



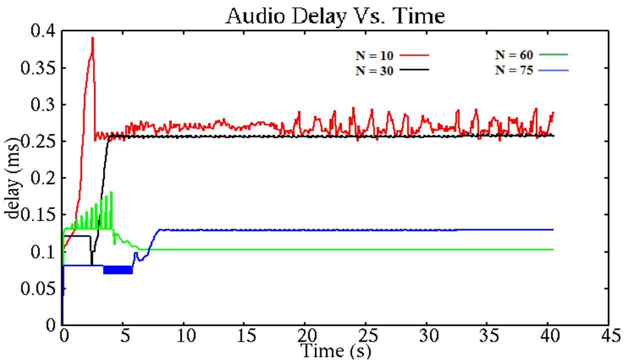
a) RED algorithm.



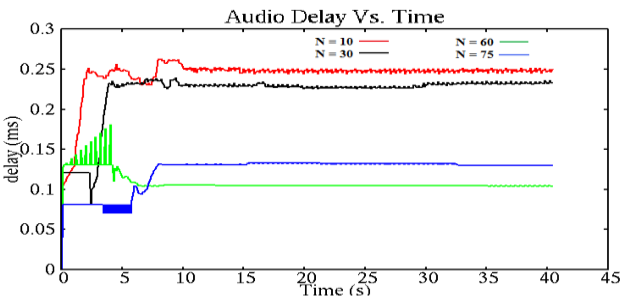
b) DQRED algorithm.

Figure 6. Delay of video traffic at different values of N.

Figure 7 shows the instantaneous packet delay for audio traffic considering the RED and the DQRED at different values of N. As the number of users increased, the delay of audio traffic is decreased, as shown in Figure 7. From the figure, considering the proposed DQRED, instantaneous delay of the audio traffic is improved compared to the normal RED.



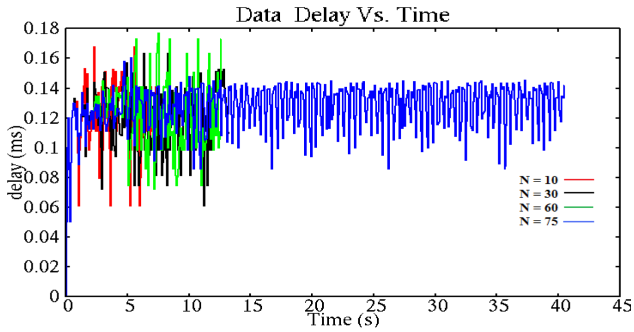
a) DQRED algorithm.



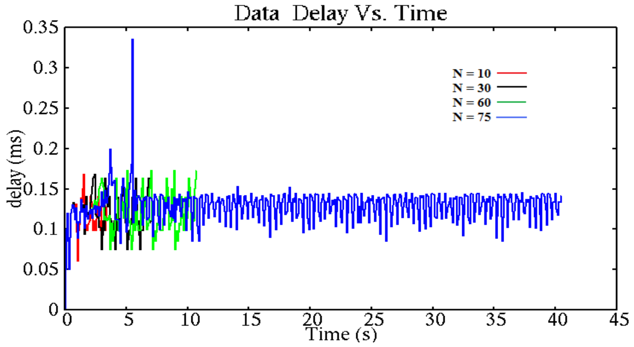
b) RED algorithm.

Figure 7. Delay of audio traffic at different values of N using RED & DQRED algorithms.

Figure 8 shows the instantaneous packet delay for data traffic considering the RED and the DQRED at different values of N. From Figure 8, for a large value of N, the packet delay of RED and the proposal DQRED are nearly identical. This means that the proposed DQRED improves the performance of video multimedia traffic applications without affecting the performance of the TCP-based traffic applications.



a) RED algorithm.



b) DQRED algorithm.

Figure 8. Delay of data traffic at different N values.

4.2.3. Effects of using DQRED on Packet Losses

Figure 9 shows the packet losses for video traffic considering the RED and the DQRED at different values of N. From the figure, the video packet losses are highly reduced with the DQRED compared with that of the RED.

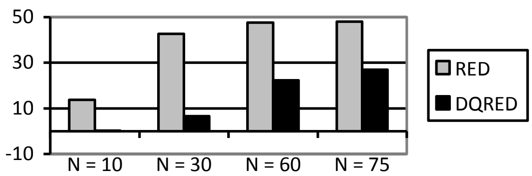


Figure 9. Video average packet losses at different N values using RED and DQRED algorithms.

Figure 10 shows the packet losses for audio traffic considering the RED and the DQRED at different values of N. From the figure, there is a significant increase in the audio packet losses with the DQRED compared to the normal RED. It is clear also that with the DQRED, the audio packet losses is nearly stable at different load traffic (except for N=30).

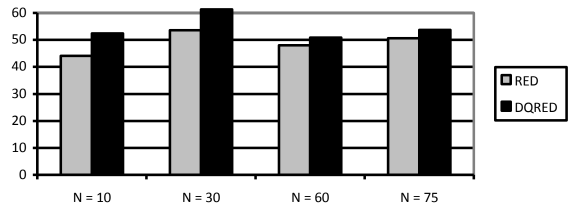


Figure 10. Audio packet losses by RED and DQRED algorithm.

Figure 11 shows the packet losses for data traffic considering the RED and the DQRED at different values of N. The figure indicates the effects of using the proposed DQRED on data traffic losses compared to the RED algorithm. It is clear that, for light traffic load, DQRED provide high data packet losses compared to RED, but with heavy traffic load (at N = 75), the performance of DQRED near that of RED.

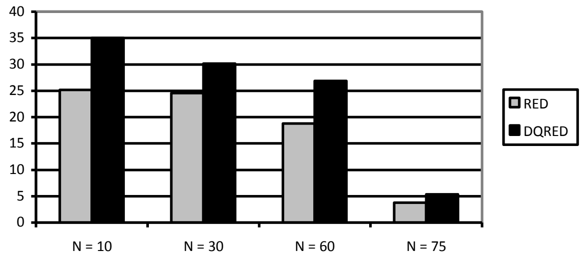


Figure 11. Data losses at different values of N.

4.3. Second Scenario: Real Network Topology

In this section, a realistic topology is used to test the performance of the proposed strategy. The AT&T real network topology, shown in Figure 12, is created by using the network topology generator GT-ITM [12]. The topology contains 166 nodes and 189 links with 75 input traffic sources; 30 FTP data applications, 20 video traffic, and 25 audio traffic applications. The simulation time is 40 seconds. In this simulation, the proposed algorithm is compared with two AQM algorithms:

- RED: Normal RED.
- RQRED: RED with three queues & static priority scheduling 3:2:1 for video, audio and data respectively.

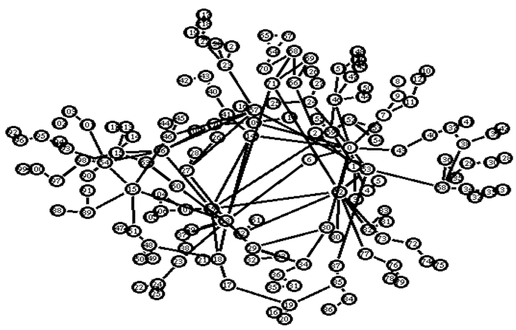
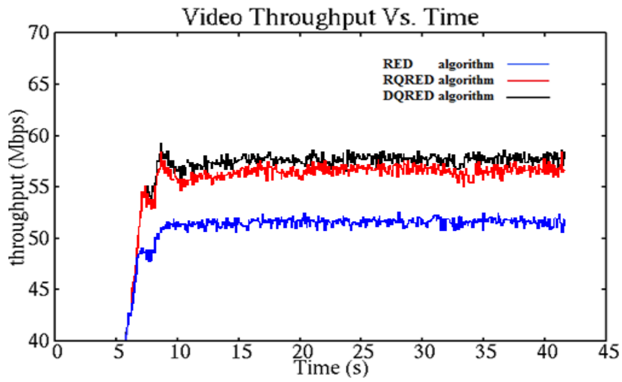


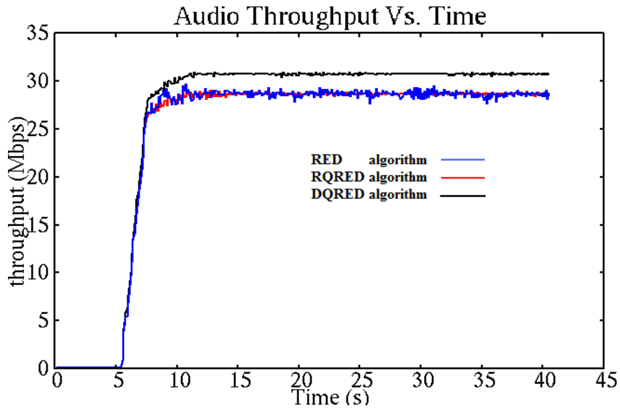
Figure 12. AT&T network topology.

4.3.1. Effects of using DQRED on Throughput

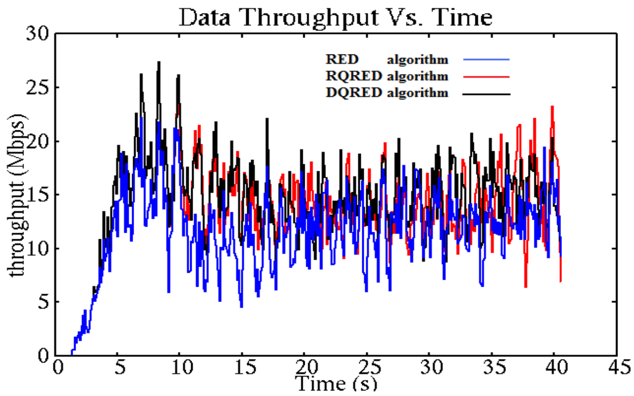
Figure 13 shows the average throughput of different traffics considering the proposed DQRED and other algorithms. The results show that the DQRED algorithm provides the highest throughput for the video and audio traffics, in addition it gives a good performance for the data traffic.



a) Video traffic throughput.



b) Audio traffic throughput.

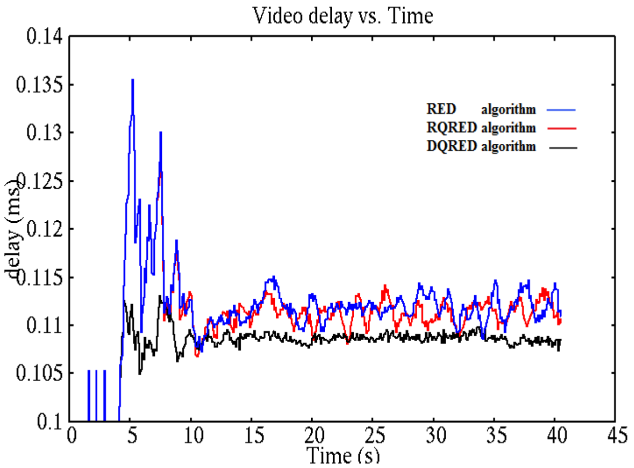


c) Data traffic throughput.

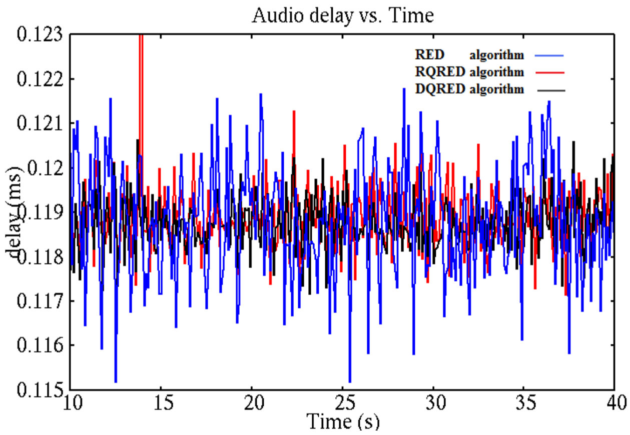
Figure 13. Average throughput of different traffic flows.

4.3.2. Effects of using DQRED on Packet Delay

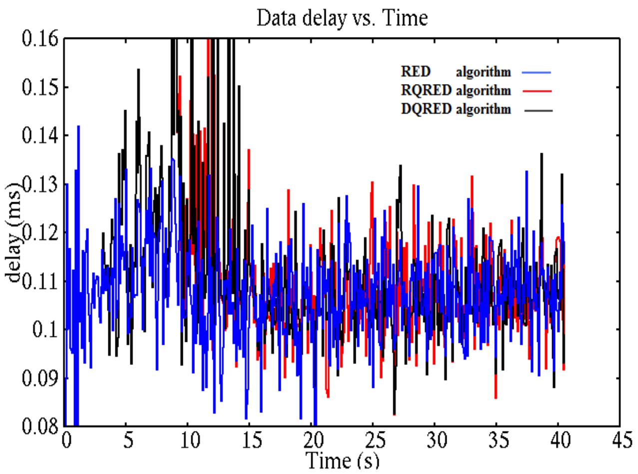
Figure 14 shows the instantaneous delay of different traffics considering the proposed DQRED and other algorithms. The figure indicates that the delays elapsed by the different algorithms are nearly identical for data packets but DQRED algorithm provides the lowest delay for the video traffic, this is the main purpose of our proposal. For audio traffic, DQRED gives the lowest delay variation (jitter) as shown in Figure 14-b.



a) Delay of video traffic.



b) Delay of audio traffic.

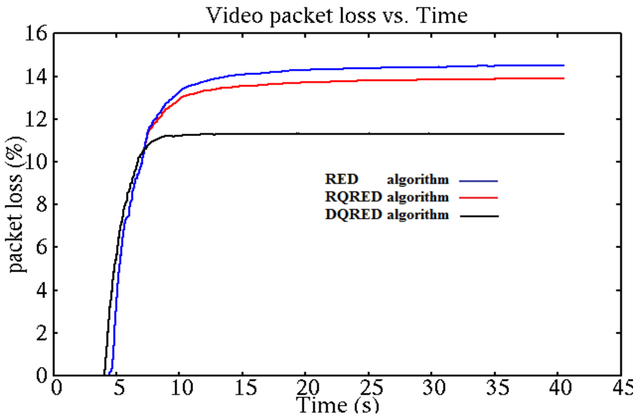


c) Delay of data traffic.

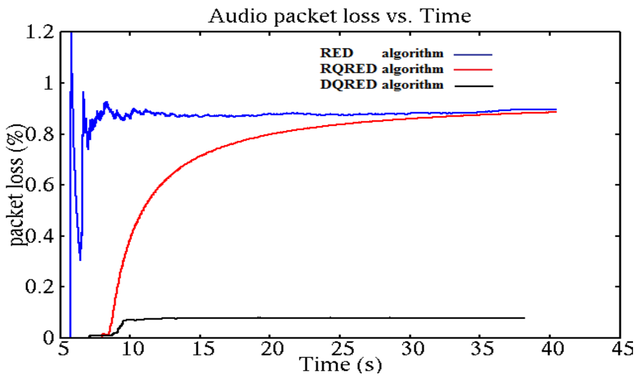
Figure 14. Packet delay of different traffic flows.

4.3.3. Effects of using DQRED on Packet Losses

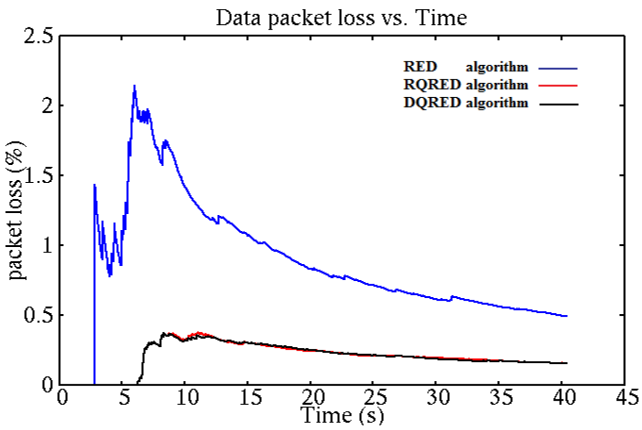
Figure 15 shows the packet loss of different traffics considering the proposed DQRED and other algorithms. From Figure 15, the packet drop ratio is reduced for the different traffic flows when using the proposed DQRED for the audio traffic. The figure also shows that the improvement of the network drop ratio when the DQRED is used compared to RED algorithm.



a) Packet loss of video traffic.



b) Packet loss of audio traffic.

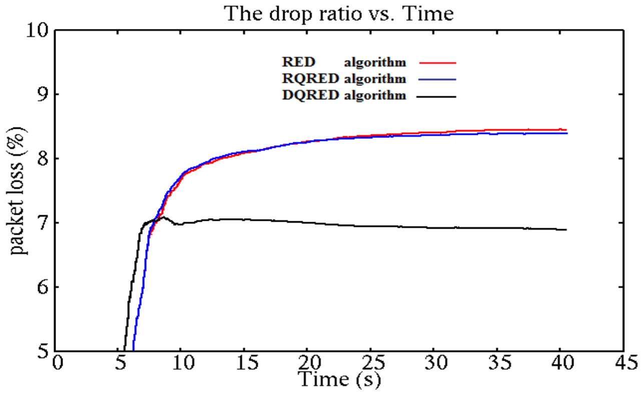


c) Packet loss of data traffic.

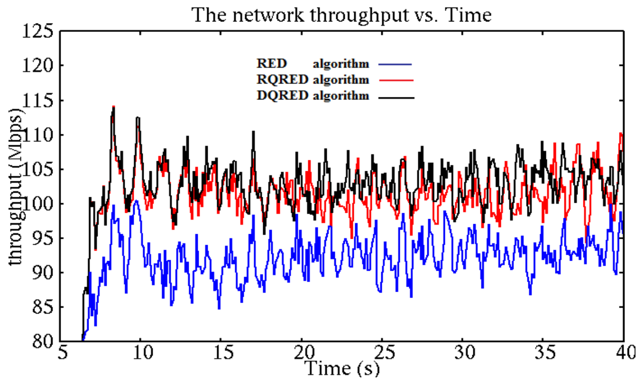
Figure 15. The packet loss of different traffic flows.

4.3.4. Effects of using DQRED on Net Performance

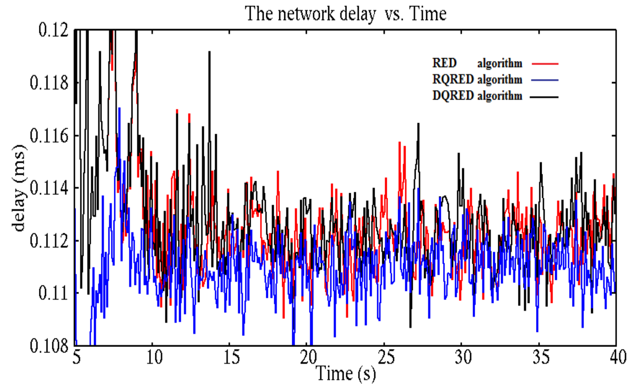
The results illustrated in Figure 16 indicate the results for the network throughput, delay, and drop ratio. From the figures, the proposed DQRED gives a slight improve in the network performance compared to the other algorithms.



a) network packet losses.



b) the network throughput.



c) the network delay.

Figure 16. Network performance with RED, RQRED and DQRED.

5. Conclusions

This paper presented a new RED-based algorithm, called Dynamic Queue RED (DQRED), to guarantee the required QoS of different traffics. In the proposed DQRED algorithm, three queues are used at the network router; one queue for each traffic type (video, audio, and data), and the queued packets are scheduled dynamically based on the number of packets of each class type. The proposed DQRED algorithm is evaluated and compared with both the normal RED and the most recent algorithms by using the network simulator NS-2. The results show that the DQRED improves the network performance in terms of throughput, delay, and the packet loss. It provides better QoS for the multimedia traffic applications without affecting the performance of TCP-based applications. An important question is how AQM algorithm can achieve low delay, low packet loss, fair buffer allocation and high throughput simultaneously; the answer will be a good challenge for future work.

References

[1] Abdulkareem M., Akil K., Kalakech A., and Kadry S., "EFRED: Enhancement of Fair Random Early Detection Algorithm," *Communications, Network and System Sciences*, vol. 8, no. 7, pp. 282-294, 2015.

[2] Adams R., "Active Queue Management: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1425-1476, 2013.

[3] Anjum F. and Tassiulas L., "Balanced-RED: An Algorithm to Achieve Fairness in the Internet," *in Proceedings of INFOCOM 8th Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, pp. 1-30, 1999.

[4] Attiya G. and El-Khobby H., "Improving Internet Quality of Service through Active Queue Management in Routers," *International Journal of Computer Science Issues*, vol. 9, no. 2, pp. 279-286, 2012.

[5] Chitra K. and Padamavathi G., "Classification and Performance of AQM-Based Schemes for Congestion Avoidance," *International Journal of Computer Science and Information Security*, vol. 8, no. 1, pp. 331-340, 2010.

[6] Christiansen M., Jeffay K., Ott D., and Smith D., "Tuning RED for Web Traffic," *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 249-264, 2001.

[7] Chung J. and Claypool M., "Dynamic-CBT-Better Performing Active Queue Management for Multimedia Networking," *in Proceedings of in International Workshop on Network and Operating System Support for Digital Audio and Video*, Chapel Hill, pp. 1-10, 2000.

[8] Feng W., Kandlur D., Saha D., and Shin D., BLUE: A New Class of Active Queue Management Algorithms, Technical Report CSE-TR-387-99, University of Michigan, 1999.

[9] Flod S., Gummadi R., and Shenker S., "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management," *in Proceedings of Floyd Adaptive RA*, pp. 1-12, 2001.

[10] Floyd S. and Jacobson V., "Link-Sharing and Resource Management Models for Packet Networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 4, pp. 365-386, 1995.

[11] Ghodsi A., Sekar V., Zaharia M., and Stoica I., "Multi-Resource Fair Queueing for Packet Processing," *in Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for*

Computer Communication, Helsinki, pp. 1-12, 2012.

[12] Heckmann O., Piringer M., Schmitt J., and Steinmetz R., How to use Topology Generators to Create Realistic Topologies, KOM Technical Report, Darmstadt University of Technology, 2002.

[13] Ismail A., EL-Sayed A., Elsaghir Z., and Morsi I., "Enhanced Random Early Detection (ENRED)," *International Journal of Computer Applications*, vol. 92, no. 9, pp. 20-24, 2014.

[14] Jacobson V. and Karels M., "Congestion Avoidance and Control," *in Proceedings of Symposium Proceedings on Communications Architectures and Protocols*, vol. 18, no. 4, pp. 314-329, 1988.

[15] Kim W. and Lee B., "FRED- Fair Random Early Detection Algorithm for TCP over ATM Network," *Electronics Letters*, vol. 34, no. 2, pp. 152-154, 1998.

[16] McCanne S. and Floyd S., "ns Network Simulator," http://www.isi.edu/nsnam/ns, Last Visited, 2015.

[17] Meckenney P., "Stochastic Fair Queuing," *in Proceedings of 9th Annual Joint Conference of the IEEE Computer and Communication Societies. The Multiple Facets of Integration*, San Francisco, pp. 733-740, 1990.

[18] Ott T., Lakshman T., and Wong L., "SRED: Stabilized RED," *in Proceedings of 8th Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, pp. 1346-1355, 1999.

[19] Rajeswari S. and Venkataramani Y., "Congestion Control and QOS Improvement for AEERG Protocol in MANET," *International Journal on AdHoc Networking Systems*, vol. 2, no. 1, pp. 13-21, 2012.

[20] Seidi M. and Hmadi A., "Enhancing QoS by Using Weighted Fair Queuing Techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 10, pp. 159-163, 2012.

[21] Shreedhar M. and Varghese G., "Efficient Fair Queuing Using Deficit Round Robin," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 375-385, 1996.

[22] Sivasubramaniam N. and Senniappan P., "Enhanced Core Stateless Fair Queuing with Multiple Queue Priority Scheduler," *The International Arab Journal of Information Technology*, vol. 11, no. 2, pp. 159-167, 2014.

[23] Wang H., Liao C., and Tian Z., "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.

[24] Zhou K., Yeung K., and Li V., "Nonlinear RED: A Simple Yet Efficient Active Queue Management Scheme," *Computer Networks*, vol. 50, no. 18, pp. 3784-3794, 2006.

Hanaa Mohammed graduated in 2000, from the Department of Electronics and Electrical Communications Engineering, Tanta University, Egypt, and she prepare 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.

Gamal Attiya graduated in 1993 and obtained his M.Sc. degree in computer science and engineering from Menoufia University, Egypt, in 1999. He received 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.

Sami El-dolil received the B. Sc. and M. Sc. degrees in electronic engineering from Menoufia University, Menouf, Egypt, in 1977and 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 the 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 network spectrum management, planning and optimization.