DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

Developed the EX-RED Algorithm to Find Best Parameters Based on Load Connection Factor

Amar A. Mahawish[1], Hassan Jaleel Hassan[2]
[1]*Computer Engineering Department, College of Engineering, Al-Iraqia University, Baghdad, Iraq*
[2]*Computer Engineering Department, University of Technology, Baghdad, Iraq*
[1]*ce.19.17@grad.uotechnology.edu.iq,* [2]*hassan.j.hassan@uotechnology.edu.iq*

***Abstract**— The performance of the Internet is significantly impacted by network congestion. Because of the internet's current rapid growth, congestion could increase and cause more packets to be dropped. The Transmission Control Protocol (TCP) connection is used as the reliable transmission of packets which has a Drop Tail (DT) mechanism since congestion DT signals occur only when the queue has become full. The Random Early Detection algorithm mechanism can be developed as congestion control which eliminates the drawback of a tail drop queue full. The Red algorithm also has problems dealing with different numbers of connections due to fix parameter tuning. In this study, the EXhaustive search used with the RED algorithm to develop the EX-RED algorithm. Based on network performance metrics such as packet drop, delay, and throughput, the developed algorithm adjusts the default RED parameter to find the best of them. When the number of TCP connections changes, the exhaustive search used systematically enumerates all possible states of the parameter. The simulation results showed that the EXRED improved performance of the network as compared with the other five algorithms* (GRED, RED, ARED, NLRED, and NLGRED) *by decreasing delay and dropped packets and increasing packet throughput.*

***Index Terms**— AQM, Congestion Control, Exhaustive search, RED.*

I. INTRODUCTION

Congestion control is more attractive to researchers because of the fast network and the quick increase in user numbers. The throughput is increased, the packet drop rate is decreased, and the overload in the router queue is managed by the congestion control algorithm to improve the Quality of Service (QoS). The router in the network is equipped with a limited buffer, and congestion fills it up, which causes dropping packets. The Slow Startup phase, which is in charge of finding a path between two end points' capacity by exponentially increasing the rate, and the Congestion Avoidance phase, which regulates the sending rate with a linearly increasing rate, are the two main phases in the TCP protocol that deal with congestion [1]. The Drop Tail DT mechanism, which drops packets only when the buffer is full, is the standard method used by TCP to handle congestion. But active queue management (AQM) [2] was created as a computer networking approach to control congestion and the amount of data that may be transmitted. The AQM techniques are essentially used to eliminate the drawback of the DT mechanism so the AQM algorithm is used as congestion control by managing the size of the router queue and preventing the queue to become full, causing a delay and packets drop. The [3] shows many algorithms implement the AQM technique and these algorithms are divided based on queue length, packet delay, or both.

Received 13/May/2023; Accepted 26/June/2023

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

Random Early Detection (RED) [4] is as default and oldest algorithm to implement the AQM technique. The RED algorithm is designed to manage and monitor the buffer size in network devices (router and switch). The RED algorithm is used to eliminate the DT mechanism from TCP data flows by computing the average queue length avg and comparing it with a predetermined threshold to determine the likelihood of packet drops. The RED parameters are the minimum threshold min_{th}, maximum threshold max_{th}, and Queue weight w_q to compute probability drop P_d. Another algorithm implements the AQM technique is Adoptive Random Early Detection ARED [5] developed to solve the fixed parameter value in the RED algorithm such as minimum threshold, maximum threshold, and probability drop by adopting the autotune these parameters to reduce drop packets. While Gentle Random Early Detection (GRED) [6, 7] employs a gentle parameter to enhance the RED algorithm reaction by doubling the range of the maxth, which results in a slower rate change in probability drop than the default RED algorithm response. NonLinear Random Early Detection (NLRED) algorithm [8] was designed to calculate the probability drop exponentially unlike the RED algorithm which determines the probability drop in linear increment, so, this algorithm will be useful when traffic load increases exponentially. The Reconfigurable NonLinear Gentle Random Early Detection (RNLGRED) [9] is similar to NLRED along with a feature of GRED that duplicates the value of maxth.

Today, Internet of Things (IoT) applications and web traffic objects both make extensive use of short TCP flows. These kinds of traffic operate by alternating between active and inactive periods of transmitting, which indicates that they belong to the Slow Startup phase and are difficult for the AQM algorithms to handle. The ON-OFF application was utilized in this work to reflect the Slow Startup flow rates in the TCP flow rate connection [10].

In order to solve the issues in RED, ARED, GRED, NLRED, and GNLRED the EXhaustive search [11] with the RED algorithm was used to find the optimal parameters. This study will take into account multiple inputs and multiple outputs, where there are several input variables and numerous output variables. The RED parameter such as min_{th}, max_{th}, sent DataRate and Queue weight are considered as input variables, while the Delay, Drop packets, and throughput as output variables. The EX-RED algorithm was used to find optimal values for input variables that give the optimal output. So, the analysis and comparison of the suggested algorithm with other algorithms depend on three parameters, namely: the number of drop packets, throughput, and delay.

The EXRED, as a suggested algorithm in this study, gives high performance in network metrics by reducing the delay and drop packets while increasing the throughput of packets. Section II presents the related work and previous study. The suggested algorithm design, the methodology used to create it, and the implementation of the RED algorithm using exhaustive search are all shown in Section III. Section IV discusses the result of the new algorithm and analysis it with other previous work.

II. RELATED WORKS

The RED algorithm was designed by Floyd and Jacobson (1993) to implement the AQM technique [4] to control the congestion of packets in the router’s queue. Many researchers were interested in congestion control that depends on the RED algorithm. The GRED algorithm [6] duplicates the range of $max_{th}$for smooth drop probability . The Dynamic GRED “DGRED” [12] same as GRED by having $doublemax_{th}$ but provides a faster response to congestion events and enhances the network's performance when using a

Received 13/May/2023; Accepted 26/June/2023
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

different type of service flow. The DGRED depends on the three-state Markov Modulated Bernoulli arrival process (MMBP-3) to support three types of traffic flow by performing correlation among these network traffics. An extension of DGRED is Stabilized Dynamic GRED “SDGRED” [13, 14]. The Improved Random Early Detection I-RED algorithm [15] uses linear and nonlinear probability dropping packets. The I-RED modifies the default RED by dividing the threshold into two sections, which means three thresholds use minth, maxth, and between them to distinguish between two types of load traffic (light and heavy load). The I-RED has two ways to compute probability drop, each one for each section. Enhanced AGRED “EAGRED” [16] was proposed to improve the response of the router’s buffer to drop packets as well as this algorithm gives better performance by reducing delay and packet loss. The EAGRED, unlike the AGRED and GRED, when using the fixed value of queue weight (e.g., 0.002) to calculate the avg which may lead delay in congestion response. The Weighted Random Early Detection “WRED” [17] was designed to drop packets based on traffic flow weight or priority in addition to multiple virtual queues and queue threshold for each virtual queue. The CISCO routing platforms support the WRED algorithm in recent series of products [18]. In [19] it was used Machine Learning techniques in the AQM algorithm with explicit congestion notification (ECN feature available in TCP header, the artificial neural network was used to predict the occurrence of congestion while the reinforcement was used to tune the AQM parameters. This algorithm was utilized to reduce the delay and to increase the throughput. In [20] the feedforward neural network with AQM FFNN-AQM technique was developed to manage congestion in router queue. The FFNN-AQM algorithm tunes the neuron's parameters based on time varying environment so the algorithm achieved a small settling time by using an adaptive neuron to find the future value of the present queue length. The [21] implement the AQM algorithm with PID and fuzzy logic controller and also used social spider optimization to tune the AQM parameters in order to reduce the error of queue size. All these studies focus on adjusting parameters to find the optimal result in the TCP congestion avoidance phase, The researcher in [22] focuses on the slow startup phase of TCP flow rate by finding the optimal queue weight parameter of AQM using the Markov Decision Process method. also in this study, congestion control occurs during the TCP slow start phase to handle congestion caused by the rapidly changing exponential increase in the data load factor.

III. DESIGN AND METHODOLOGY

A. AQM Technique

Active Queue Management is a technique used in the network devices queue (router queue) to reduce the congestion in the router queue to enhance the performance of data transmission on the internet. The RED algorithm is a simple one that implements the AQM feature by using random probability drop packets and monitoring the value of the current average queue length avg which can be calculated based on equation 1. To overcome the drawback of the drop tail mechanism in TCP flows, the packets in RED were dropped before the queue reached capacity. The RED parameters are the minimum threshold min_{th}, maximum threshold max_{th}, and Queue weight w_q to compute probability drop P_d. The probabilistic mark/drop was determined for each incoming packet, as shown in *Fig. 1* when the avg is less than the min_{th}, the P_d is null so there are no packets dropped. When the avg passed the max_{th}, the probability set to one, meaning that all incoming packets will be dropped. When the avg was between min_{th} and max_{th}, the probability of drop packets was computed by equation 2.

Received 13/May/2023; Accepted 26/June/2023
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

$$avg = (1 - w_q)avg + w_q q \quad (1)$$

$$if\ min_{th} < avg \leq max_{th} \quad then\ the$$

$$P_d = max_p \frac{avg - min_{th}}{max_{th} - min_{th}} \quad (2)$$

Where avg refers to average queue length, w_q refer to queue weight, q for queue length, min_{th} for minimum threshold, max_{th} for maximum threshold and the P_d for dropping probability.

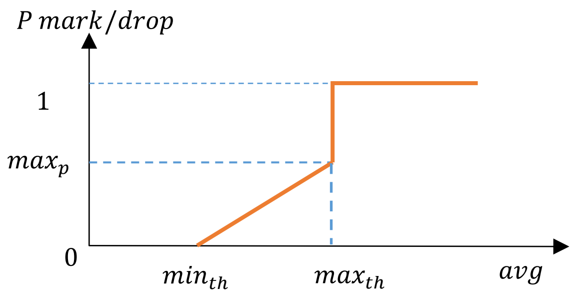


FIG. 1. RED ALGORITHM.

B. The Proposed Algorithm

Exhaustive search or brute force search is an algorithm that consists of systematically testing all possible solutions or combinations of solutions to a problem to find the optimal one. the exhaustive search is used to find the best parameters (min_{th}, max_{th}, w_q and D_r) to find the best value for Drop packets, Throughput, and Delay as shown in *Fig. 2*.

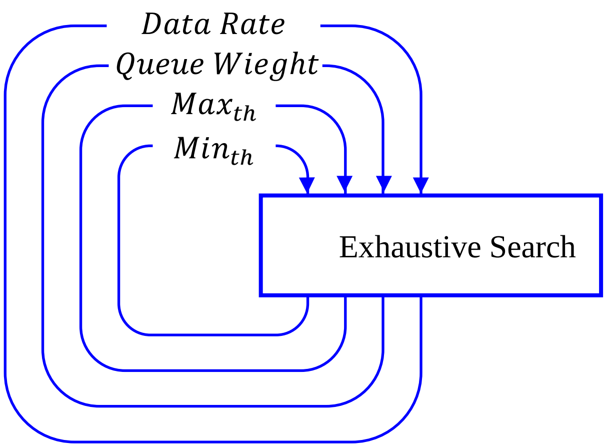


FIG. 2. THE EXHAUSTIVE SEARCH OPERATION.

In this study, the designed algorithm for congestion control depends on the RED algorithm by utilizing the feature of the exhaustive search. By using Network Simulator 3 "NS3", it was implemented the developed algorithm to find the optimal value of Drop packets, Throughput, and Delay. The pseudocode algorithm of the proposed algorithm is shown below.

Received 13/May/2023; Accepted 26/June/2023

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

Algorithm 1:

The pseudocode of the proposed algorithm 1

1. Set Parameters (packetSize, dataRate, min, and max threshold)
2. //the Exhaustive search processes
3. For: NumberOfNode → 5 to 200 step 5
4. For: DataRate → 5 to 15 step 5
5. For minth → 3 to 7
6. For maxth → 13 to 17
7. For qw → 0.001 to 0.009 step 0.001
8. find the maximum throughput
9. find the minimum of Delay and Drop
10. for each packet arrival
11. Calculate the average queue (avg)
12. $avg = (1 - w_q)avg + w_q q$
13. Calculate drop probability

C. The Implantation and Network Design

The Dumbbell network topology was designed as shown in *Fig. 3*.a by using NS3 with an ON-OFF application to implement and compare the proposed algorithm with other algorithms. *Fig. 3*.b shows the bottleneck between router 0 and router 1 and how congestion occurred in that link after 20 ms of simulation running time. The number of node connections to router 0 and router 1 will reflect the load factor that can increase in each run.

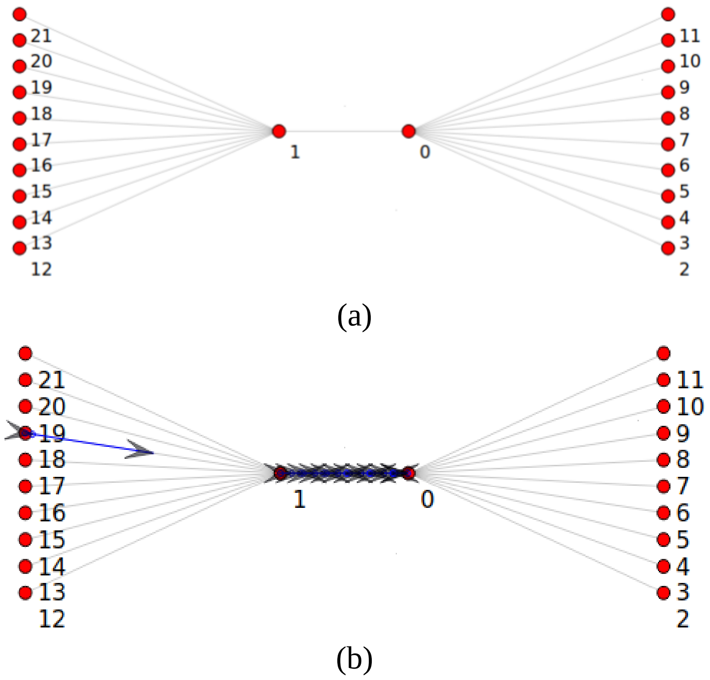


FIG. 3. NS3 POINT-TO-POINT DUMBBELL TOPOLOGY (A) BEFORE STARTING SIMULATION AND (B) AFTER 1 SEC OF RUN TIME.

Algorithm 1 shows the range of each parameter of the EX-RED algorithm, the load factor (number of connected nodes) in this study uses a different level of load vector that was divided into three groups (light, moderate, and heavy loads) as shown in Table I. While another parameter of the RED algorithm will change based on Table II.

Received 13/May/2023; Accepted 26/June/2023

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

TABLE I. NAME OF EACH RANGE OF NODES CONNECTION

Number of connection nodes	Traffic load factor
5-50	light load
55-100	moderate load
105-150	heavy load

TABLE II. THE EX-RED PARAMETER VALUES RANGE

Parameter name	Value Range	steps
min_{th}	3-6	1
max_{th}	13-16	1
q_w	0.001-0.009	0.001
D_r	5-15	5

IV. SIMULATION RESULTS AND DISCUSSION

The results of this study were obtained by running EX-RED in NS3 with a variety of samples. The samples were obtained by changing the RED algorithm's parameters based on Tables I and II, as shown in *Fig. 4,* and applying an exhaustive search to determine the parameter values that would produce the best throughput, delay, and packet drop results. These three parameters were considered as metrics to measure the performance in networks.

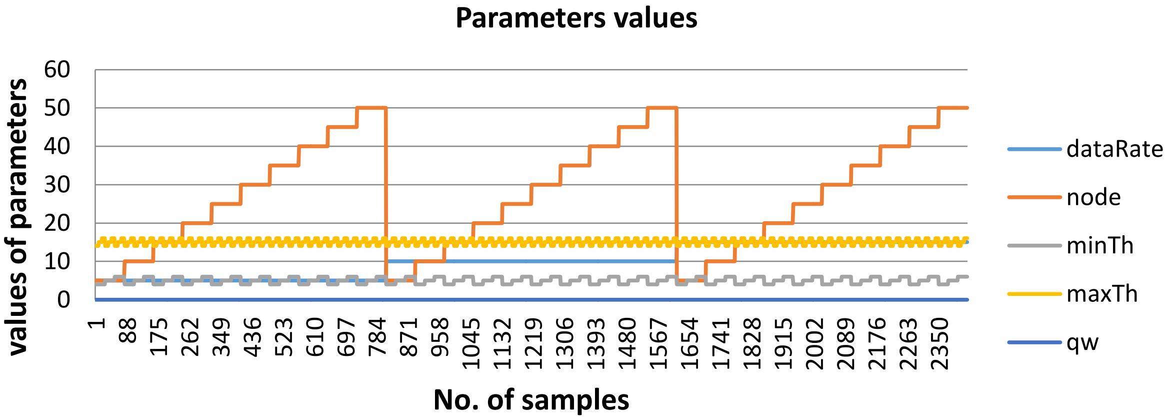


FIG. 4. PARAMETERS VALUES FOR ALL PARAMETERS.

Where the value of the data rate changes from 5 to 15 with 5 steps in each round. The number of nodes changes from 5 to 150 with 5 steps in each round. The min_{th} 3 to 6 with steps 1 for each round. The max_{th} rang from 13 to 16 with 1 step in each round. And last parameter the q_w with rang o.001 to 0.009 with steps 0.001 in each round.

A. Light Load Connection

In this subsection, the number of nodes connected in the network can vary from 5 to 50 nodes, increasing by 5 nodes each time. The simulation execution shows the throughput, drop packets, and delay for each simulation run as shown in *Fig. 5*. This figure shows the overall change can be represented when the simulation run and the suggested algorithm choose the best value of throughput, drop, and delay in every load traffic change (number of nodes) during light load traffic.

Received 13/May/2023; Accepted 26/June/2023

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

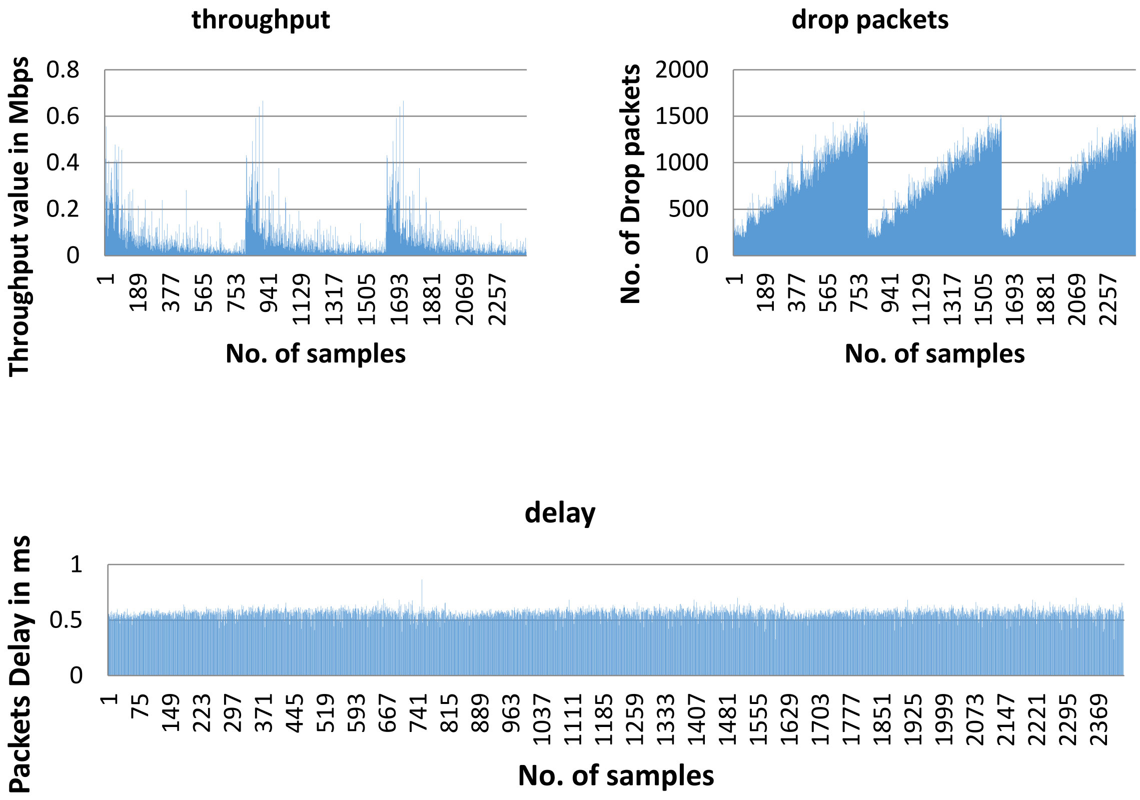
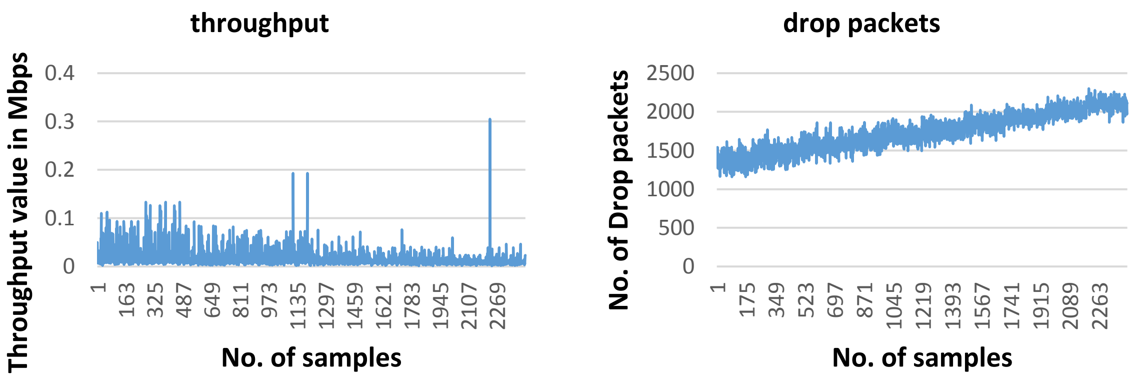


FIG. 5. THE LIGHT LOAD SIMULATION RESULT FOR THROUGHPUT, DROP PACKETS, AND DELAY.

B. Moderate Load Connection

Similar to the previous subsection, but with a range of 55 to 100 nodes and an increment of 5 nodes every step in the network's connected nodes. The simulation execution shows the throughput, drop packets, and delay for each simulation run as shown in *Fig. 6*. This figure shows the overall change can be represented when the simulation run and the suggested algorithm choose the best value of throughput, drop, and delay in every load traffic change (number of nodes) during moderate load traffic.



Received 13/May/2023; Accepted 26/June/2023
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

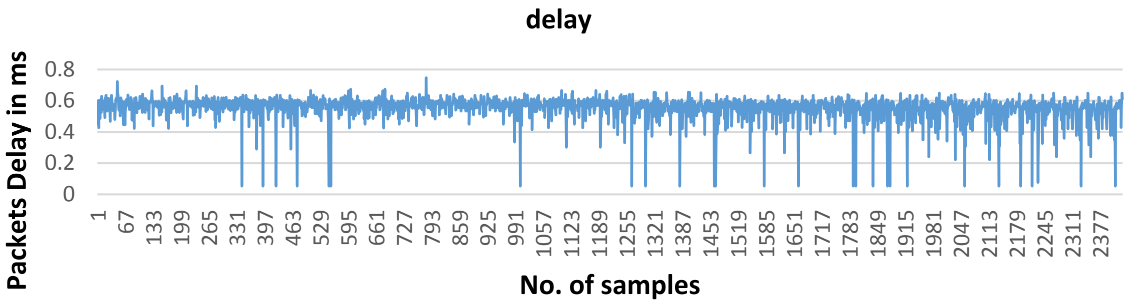


FIG. 6. THE MODERATE LOAD SIMULATION RESULT FOR THROUGHPUT, DROP PACKETS, AND DELAY.

C. Heavy Load Connection

Similar to the previous subsection, but with a range of 55 to 100 nodes and an increment of 5 nodes every step in the network's connected nodes. The simulation execution shows the throughput, drop packets, and delay for each simulation run as shown in *Fig. 7*. This figure shows the overall change can be represented when the simulation run and the suggested algorithm choose the best value of throughput, drop, and delay in every load traffic change (number of nodes) during heavy load traffic.

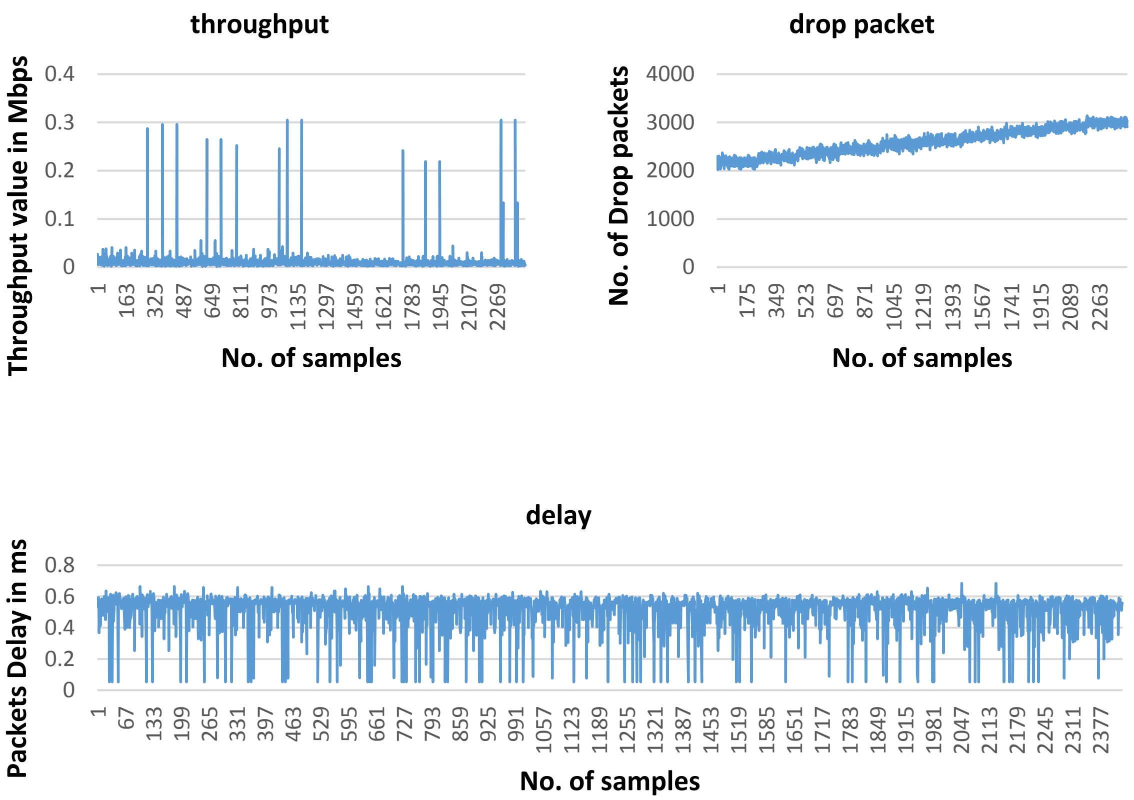


FIG. 7. THE HEAVY LOAD SIMULATION RESULT FOR THROUGHPUT, DROP PACKETS, AND DELAY.

The suggested algorithm performance has been compared with five other algorithms based on delay, throughput, and drop packets. As shown in *Fig. 8*. The figure shows the

Received 13/May/2023; Accepted 26/June/2023

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

best value of overall parameters (data rate, min_{th}, max_{th}, and queue weight). The new algorithm shows the best performance when the number of nodes changes from 5 to 150 based on three clusters of the traffic load (light, moderate and heavy load). However, the suggested algorithm improve the delay of delivered packets of EX-RED that presented a low delay compared to GRED, RED, ARED, NLRED, and NLGRED algorithms.

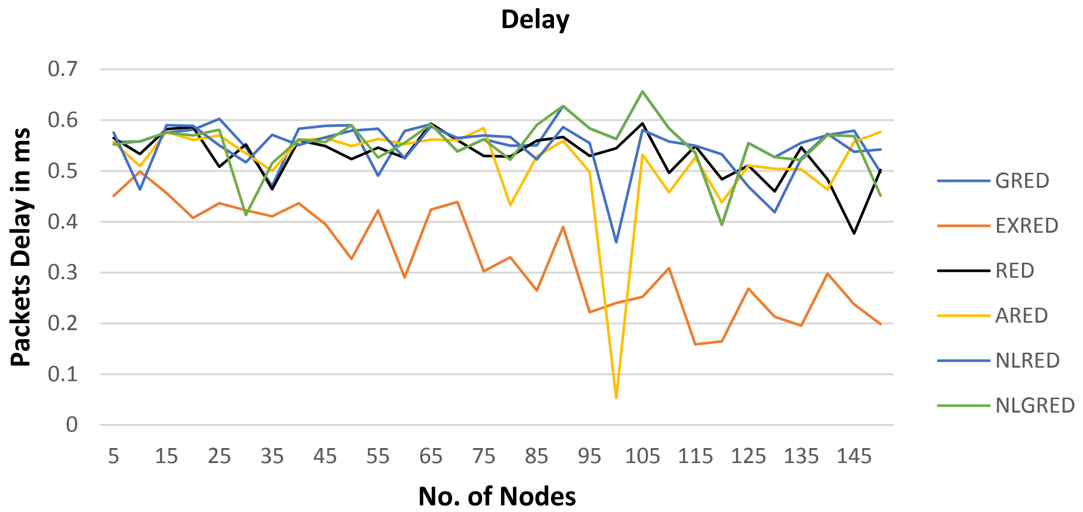


FIG. 8. THE EX-RED DELAY PERFORMANCE WITH OTHER ALGORITHMS.

The *Fig. 9.* shows the best value of overall parameters (data rate, min_{th}, max_{th}, and queue weight). The new algorithm shows the best performance when the number of nodes changes from 5 to 150 based on three clusters of the traffic load (light, moderate and heavy load). So, as the figure can conclude the EXRED gives higher throughput packets than the other five algorithms, which means that the suggested algorithm has better results as shown in figure below.

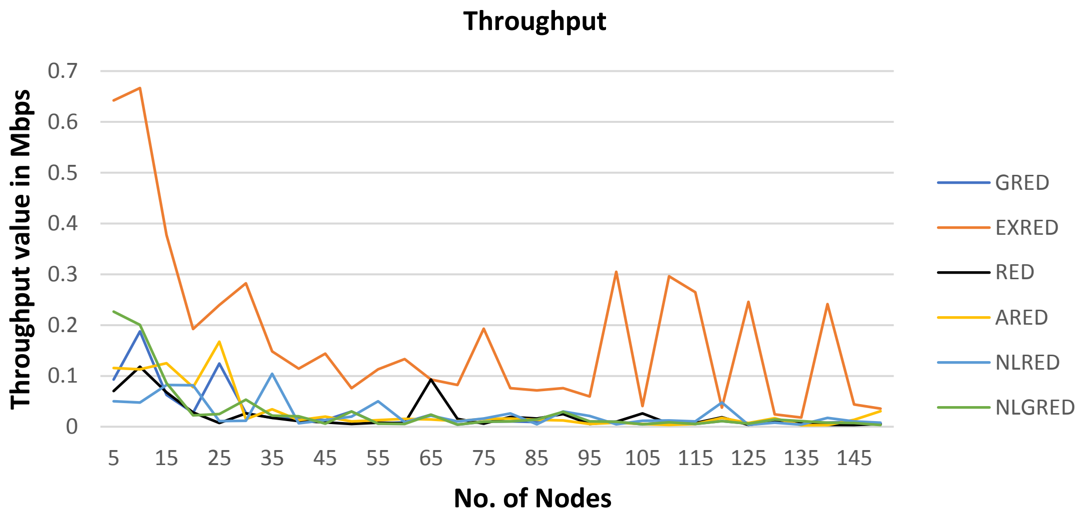


FIG. 9. THE EX-RED THROUGHPUT PERFORMANCE WITH OTHER ALGORITHMS.

Received 13/May/2023; Accepted 26/June/2023
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

While the *Fig. 10* shows the best value of overall parameters (data rate, min_{th}, max_{th}, and queue weight). The new algorithm shows the best performance when the number of nodes changes from 5 to 150 based on three clusters of the traffic load (light, moderate and heavy load). So, the low packet drop given by the EX-RED algorithm is compared with other algorithms.

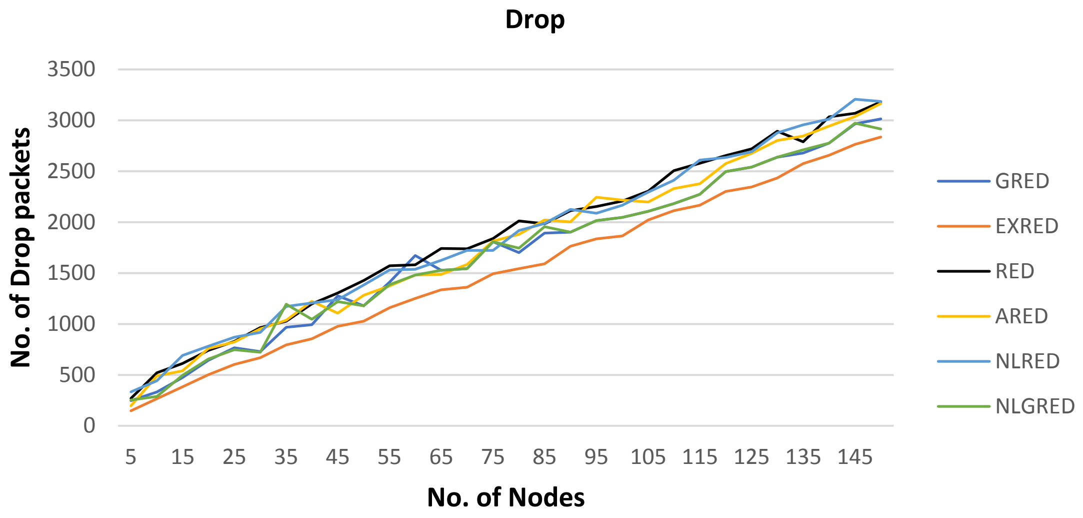


FIG. 10. THE EX-RED PACKETS DROP PERFORMANCE WITH OTHER ALGORITHMS.

V. CONCLUSIONS

The suggested EXRED algorithm has been tested based on the three most important network metrics of performance:(Delay, Throughput, and Drop packets). The EXRED algorithm compared to the other five algorithms (GRED, RED, ARED, NLRED, and NLGRED), the algorithm developed in this study yields the best results. The EXRED gives lower delay and drop packets, while in throughput metrics it gives a higher result as compared with other algorithms.

REFERENCES

[1] M. Allman, V. Paxson e E. Blanton, "TCP Congestion Control," *Network Working Group, RFC5681,* September 2009.

[2] F. Baker e G. Fairhurst, "IETF Recommendations Regarding Active Queue Management," *Internet Engineering Task Force (IETF), RFC7567,* p. 31, July 2015.

[3] Amar A. Mahawish and Hassan J. Hassan, "Survey on: A variety of AQM algorithm schemas and intelligent techniques developed for congestion control," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 23, nº 3, pp. 1419-1431, September 2021.

[4] S. Floyd e V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking,* vol. 1, nº 4, pp. 397-413, August 1993.

[5] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management," *AT&T Center for Internet Research at ICSI,* p. 12, August 2001.

[6] S. Floyd, "Recommendation on using the "gentle_" variant of RED," March 2000. [Online]. [Acesso em 2021].

[7] N. Hamadneh, M. Al-Kasassbeh, I. Obiedat e M. BaniKhalaf, "Revisiting the Gentle Parameter of the Random Early Detection (RED) for TCP Congestion Control," *Journal of Communications,* vol. 14, nº 3, p. 7, March 2019.

[8] H. Abdel-Jaber, "An Exponential Active Queue Management Method Based on Random Early Detection," *Journal of Computer Networks and Communications,* 2020.

Received 13/May/2023; Accepted 26/June/2023

DOI: https://doi.org/10.33103/uot.ijccce.23.4.8

[9] Ahmad F. AL-Allaf and A. I. A. Jabbar, “Reconfigurable Nonlinear GRED Algorithm,” *International Journal of Computing and Digital Systems,* nº 5, 2020.

[10] I. Järvinen, “Congestion Control and Active Queue Management During Flow Startup,” *Series of publications A/Department of Computer Science, University of Helsinki,* 2019.

[11] B. R. Al-Kaseem, H. S. Al-Raweshidy, Y. Al-Dunainawi and K. Banitsas, “A New Intelligent Approach for Optimizing 6LoWPAN MAC Layer Parameters,” *IEEE Access,* vol. 5, pp. 16229-16240, 2017.

[12] M. Baklizi, J. m. Ababneh, M. Abualhaj, N. Abdullah e R. Abdullah, “Markov-modulated bernoulli dynamic gentle random early detection,” *Journal of Theoretical and Applied Information Technology,* vol. 96, nº 20, October 2018.

[13] M. Baklizi, “Stabilizing Average Queue Length in Active Queue Management Method,” *IJACSA,* vol. 10, nº 3, p. 7, 2019.

[14] M. Baklizi, “Weight Queue Dynamic Active Queue Management Algorithm,” *Symmetry,* vol. 12, nº 12, p. 2077, December 2020.

[15] Samuel O. Hassan, Adewole U. Rufai, Michael O. Agbaje, “Improved random early detection congestion control algorithm for internet routers,” *ijeecs,* vol. 28, nº 1, pp. 384-395, 2022.

[16] M. Baklizi e J. Ababneh, “Performance Evaluation of the Proposed Enhanced Adaptive Gentle Random Early Detection Algorithm in Congestion Situations,” *International Journal of Current Engineering and Technology,* vol. 6, nº 5, p. 8, October 2016.

[17] R. Molenaar, “https://networklessons.com,” cisco, 2013 - 2021. [Online]. Available: https://networklessons.com/cisco/ccie-routing-switching-written/wred-weighted-random-early-detection#.

[18] CISCO, “Queue Limits and WRED,” em *QoS Modular QoS Command-Line Interface Configuration Guide, Cisco IOS XE 17*, 2020.

[19] C. A. Gomez, X. Wang and A. Shami, “Intelligent Active Queue Management Using Explicit Congestion Notification,” *2019 IEEE Global Communications Conference (GLOBECOM),* pp. 1-6, 2019.

[20] Sukant Kishoro Bisoy, Prasant Kumar Pattnaik, “An AQM Controller Based on Feed-Forward Neural Networks for Stable Internet,” *Arab J Sci Eng,* p. 3993–4004, 2018.

[21] Berbek, Mohammed I., Oglah, Ahmed A., “Fuzzy Like Pid Controller Based on Sso Design for Congestion Avoidance in Internet Router,” *IJCCCE,* vol. 21, nº 2, pp. 85-101, 2021.

[22] Amar A. Mahawish, Hassan J. Hassan, “Improving RED algorithm congestion control by using the Markov decision process,” *Scientific Reports,* vol. 12, p. 9, 2022.

Received 13/May/2023; Accepted 26/June/2023
