HRED, An Active Queue Management Approach For The NS2 Simulator

Article *in* Recent Patents on Computer Science · December 2018

DOI: 10.2174/2213275912666181205155828

4 authors:

Nabhan Hamadneh
Hashemite University

Mamoon Obiedat
Hashemite University

Ahmad Qawasmeh
Hashemite University

Mohammad Bsoul
Hashemite University

RESEARCH ARTICLE

HRED, An Active Queue Management Algorithm for TCP Congestion Control

Nabhan Hamadneh*, Mamoon Obiedat, Ahmad Qawasmeh and Mohammad Bsoul

Computer Science and Applications Dep. Hashemite University, P.O. Box 21233, Zarqa Jordan

Abstract: ***Background***: Active Queue Management (AQM) is a TCP congestion avoidance approach that predicts congestion before sources overwhelm the buffers of routers. Random Early Detection (RED) is an AQM strategy that keeps history of queue dynamics by estimating an average queue size parameter *avg* and drops packets when this average exceeds preset thresholds. The parameter configuration in RED is problematic and the performance of the whole network could be reduced due to wrong setup of these parameters. Drop probability is another parameter calculated by RED to tune the drop rate with the aggressiveness of arriving packets.

Objective: In this article, we propose an enhancement to the drop probability calculation to increase the performance of RED.

Method: This article studies the drop rate when the average queue size is at the midpoint between the minimum and maximum thresholds. The proposal suggests a nonlinear adjustment for the drop rate in this area. Hence, we call this strategy as the Half-Way RED (HRED).

Results: Our strategy is tested using the NS2 simulator and compared with some queue management strategies including RED, TD and Gentle-RED. The calculated parameters are: throughput, link utilization and packet drop rate.

Conclusion: Each performance parameter has been plotted in a separate figure; then the robustness of each strategy has been evaluated against these parameters. The results suggest that this function has enhanced the performance of RED-like strategies in controlling congestion. HRED has outperformed the strategies included in this article in terms of throughput, link utilization and packet loss rate.

Keywords: NS2, network simulation, TCP, congestion control, Active Queue Management (AQM), network performance.

1. INTRODUCTION

Current high-speed network gateways are likely to be congested due to the increased demand for limited network resources such as routers and link bandwidths. The free space of buffers in routers should be large enough to accommodate the short-term bursty traffic [1]. TCP congestion control strategies attempt to manage congestion by manipulating the congestion window size *cwnd*; which is a parameter that regulates the sending rate [2].

These algorithms adjust the sending rate upon congestion signal due to packet drop in intermediate router. Congestion signal could be a duplicate acknowledgment, time-out or Explicit Congestion Notification (ECN). TCP Tahoe, TCP Reno, New-Reno, TCP Vegas and Cubic [3] are examples of these strategies. In this paper, we will refer to these algorithms as source algorithms [4, 5].

How and when these congestion signals are sent, is controlled by the routers between the sender and receiver. These routers should run an Active Queue Management (AQM) scheme to control their buffers [6] and to stabilize the queue size [7]. Random Early Detection (RED) and its variants are classic AQMs for congestion avoidance but the default option, in many cases is known as Tail Drop [8]. In this paper, we will refer to these algorithms as AQM algorithms [9].

There is a new suggestion in current research to avoid using packet drop as a congestion signal [10, 11]. However, the majority of Active Queue Management (AQM) strategies rely on packet drop event for starting congestion reaction [12]. In this article, we consider packet loss as an indicator of congestion. We also propose a new congestion control strategy to overcome some of the drawbacks of other RED-based strategies.

*Address correspondence to this author at the Computer Science and Applications Dep. Hashemite University, P.O. Box 21233, Zarqa Jordan;
Tel: +962-5-3903333-EX: 4950; Fax: +962-5-382-6625;
E-mail: nabhan@hu.edu.jo

2213-2759/19 $58.00+.00

The rest of this article is organized as follows: Section 2 presents related work to the congestion problem of TCP networks. Section 3 presents the analysis and motivation of our proposal. Section 4 shows the simulation results and Section 5 outlines current and future works in our research.

2. RELATED WORK

Active Queue Management (AQM) is an effective approach for congestion control at routers. Accustomed AQM algorithms have been adopted by routers to maintain high network performance; examples of such strategies are the Tail Drop (TD), Random Early Detection (RED), Random Early Marking (REM). RED is one of the most popular AQM algorithms that has been extensively discussed by researchers, and Many RED's variants have been proposed in the AQM literature. RED-based strategies improve the performance by modifying the dropping function to minimize the packets loss rates.

Tail Drop (TD) is considered the simplest active queue management algorithm [13]. It uses First in First out (FIFO) queue management. When the router buffer has overloaded a packet is dropped, then a source algorithm interprets this as congestion. TCP protocol in source node reacts and reduces the sending rate by adjusting the congestion window size. This leads to the congestion collapse problem and poor fairness [9].

Fig. (**1**) shows how TD reacts to congestion. All the parameters depicted in the figure have no meanings to TD including min_{th}, max_{th}, $midpoint$ and max_p, except the q_Size parameter. We keep them in Fig. (**1**) just to compare them with the drop probability of RED in Fig. (**2**). Here, TD would not drop any packet unless the queue size reaches the available buffer limit. This is where the drop probability is suddenly incremented from *0.0* to *1.0*. Therefore, the drop probability determines the percentage of how many packets to drop out of all arriving packets. This scenario leads to a problem called the Global Synchronization [14] when all TCP sources reduce their sending rate at the same time resulting in low bandwidth utilization.

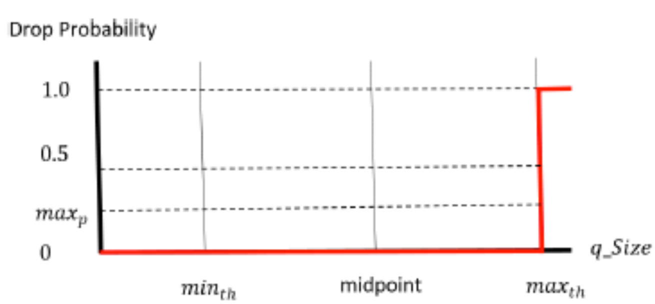

Fig. (1). Drop probability for TD.

In addition to the Global Synchronization problem, TD suffers from two other problems which are: The Lock Out and Full Queue problems. Lock out occurs when few connections monopolize the queue space. Full queue problems occur when the router keeps sending full queue signals to the sources for long period of time [13]. If TD manages to drop packets before the buffer becomes full, then the strategy overcomes all these problems. RED and its variants have fixed these drawbacks of the TD by dropping packets gradually with adjustable drop probability.

RED maintains an Exponentially Weighted Moving Average (EWMA) of the queue size on routers. Equation 1, illustrates the calculation of this average. Equations 2 and 3 illustrate how the drop rates of packets are calculated. The drop rate p_a in Eq. 3 is an accumulative linear value which is calculated using current drop probability p_b in Eq. 2. Instead of the actual queue size in TD, RED drops packets depending on the average queue size (avg). If the average queue size is between the min_{th} and max_{th} then packets are dropped with the probability p_a. IF the average is greater than or equal to the max_{th} then RED drops packets with probability *1.0*. In other words, RED will drop all arriving packets as same as TD does in case of full buffer.

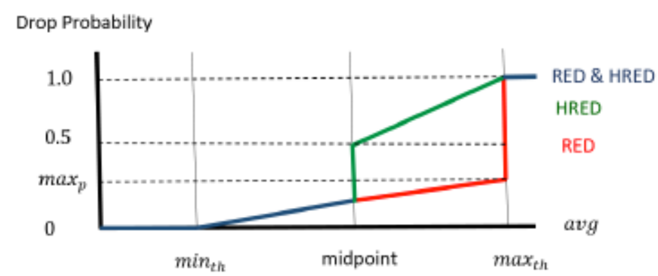

Fig. (2). Drop probability for RED and HRED.

$$avg = (1 - w_q) * avg + w_q * q \tag{1}$$

$$p_b = max_p \left(\frac{avg - min_{th}}{min_{th} - max_{th}}\right) \tag{2}$$

$$p_a = p_b \left(\frac{1}{1 - count * p_b}\right) \tag{3}$$

Where is:

avg : Average queue size.

w_q : A weight parameter, $0 \leq w_q \leq 1$.

q: The current queue sizes.

p_b: Immediately marking probability.

max_p: Maximum value of p_b.

min_{th}: Minimum threshold.

max_{th}: Maximum threshold.

p_a: Accumulative probability.

$count$: number of undraped packets since the last dropped one.

Even though RED fixed the problems of TD, their always have been mismatches between the average and actual queues. This could lead to the possibility of packet discards even when the queue size is lower than the threshold value. In addition, RED is very sensitive to parameter configuration and the performance could be reduced dramatically with wrong parameter setups. Therefore, RED-based strategies have been proposed to combat these problems. However, AQM approaches including RED are still suffering from

Table 1. Drop rates for different max_p values.

max_p	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
Midpoint: 1 out of___	50	34	25	20	17	15	13	12	10
Before max_{th} : 1 out of___	25	17	13	10	9	8	7	6	5

difficult configuration and negative impact on network utilization [15]. This research proposes a new flavor of RED and compares it with current proposed strategies through network simulation. The strategies included in this comparison are the original RED, Gentle-RED and the classical queuing algorithm Tail Drop.

RED's variants try to stabilize the queue around a target value [7, 16] and produce more throughputs [17] which can be achieved significantly using dynamic adaptation of the max_p parameter [18]. If the drop probability p_a lies between max_p and *1.0,* then this will result in oscillations of queue length which leads to an increase in the queuing delay [19]. On the other hand, the linear increment of the drop probability between the maximum and minimum thresholds- as depicted in Fig. (**2**) - causes oscillations of queue length; resulting in poor performance for the network [18, 19]. Hence, our proposal in HRED is to keep nonlinear drop probability in this period as illustrated in Fig. (**2**). HRED is an easy to deploy RED variant that keeps the queue around a target value [20]. Reference [21] is a good resource for detailed information in previous work of RED and other Active Queue Management techniques.

3. ANALYSIS OF THE MAXIMUM DROP RATE AND MOTIVATION OF HRED

RED is very sensitive to parameter configuration. The most critical parameter of RED is the maximum drop rate max_p which was suggested to take values between *0.02* and *0.1* [22]. When the average queue size is between the minimum and maximum thresholds, RED drops packets with probability p_a which is a function of the max_p; as illustrated in Equations 2 and 3. The max_p parameter determines the number of arrived packets between two drops which is called the intermarking time and denoted by X in Equation 4.

$$prob[X = n] = \frac{p_b}{1-(n-1)p_b} \prod_{i=0}^{n-2} \left(1 - \frac{p_b}{1-ip_b}\right) \quad (4)$$

From equation 4, $prob[x = n] = p_b \; for \; 1 \leq n \leq 1/p_b$ and $prob[x = n] = 0$ for $n > 1/p_b$. Thus, X would take values from the set $\{1, 2\ldots,1/p_b\}$. For example, if the max_p parameter has been assigned the value *0.02* then RED will drop *1* out of *50* arriving packets when the average queue size is between the minimum and maximum thresholds. On the other hand, if the max_p parameter has been assigned the value *0.1* then RED will continue dropping nearly *1/5th* of the arriving packets just before the average queue size hits the maximum threshold in which all arriving packets have to be dropped with probability *1.0*.

Table **1** shows a range of max_p values and the drop frequencies associated with each value for two different *avg* values. The first is the midpoint between the maximum and minimum thresholds and the second is just before the *avg* hits the maximum threshold.

RED controls congestion by reducing the queue oscillations which is governed by the drop probability. However, due to the sudden increase in the drop probability from *0.2* to *1.0* when the average queue size hits the maximum threshold, the whole performance of RED would be degraded. The network would reduce the sending rate and it might take a long period of time to regain a higher sending rate depending on the source algorithm in use. As a result, the drop rate will be increased with low throughput and link utilization. In addition, if max_p equals to *0.1* then RED will drop *1* out of *10* packets when *avg* is at the midpoint between the minimum and maximum thresholds. Our proposal is to drop *50%* of traffic at this point to handle congestion before it is out of hand, also to increase the drop probability gradually with respect to the severity of congestion; especially, when the average queue size is approaching the maximum threshold to avoid the sudden full congestion indicator. Fig. (**2**) shows the difference between normal RED strategy and our strategy in calculating the drop probability. Table **2** shows the algorithm of HRED.

Table 2. HRED's algorithm.

Preset min_{th}, max_{th}, max_p, w_q
Set $avg = 0$, $midpoint = min_{th} + (max_{th} - min_{th})/2$
For every packet arrival update avg (Eq.1)
IF ($avg \geq min_{th}$ && $avg < midpoint$) THEN
Calculate p_b(Eq. 2)
Calculate p_a(Eq. 3)
Drop arriving packets with probability p_a
ELSE IF ($avg = midpoint$) THEN
p_a=0.5
ELSE IF ($avg \geq midpoint$ && $avg < max_{th}$) THEN
Update p_a(Eq. 3)
ELSE IF ($avg \geq max_{th}$) THEN
p_a=1.0

4. SIMULATION

4.1. Simulation Scenario

Fig. (**3**) illustrates the network topology that is used to test HRED. In this topology, three sources send TCP packets using TCP-Reno. A sink immediately sends an acknowledgment packet when it receives a data packet. Arrivals of sessions follow a Poisson process.

A connection between each node and the gateway has *1Mbps* bandwidth with *4ms* delay time. The bottleneck link between the gateway and the sink has *0.1Mbps* bandwidth with *20ms* delay time for delivering the packet to the sink. Exponential distribution is used for the start time of packets transfers.

The following section describes our simulation results over the three introduced strategies with respect to network performance parameters.

4.2. Results

This section presents a comparison between the original RED, Gentle-RED, the classical Tail Drop and the proposed HRED strategy. Figs. (**4-6**) show the throughput, link utilization and packet loss respectively.

Fig. (**4**) shows that HRED produced the highest throughput throughout the simulation time while the original RED shows the lowest throughput. HRED also provided higher link utilization over the other strategies, as shown in Fig. (**5**). Fig. (**6**) shows that there are fewer oscillations in packet losses.

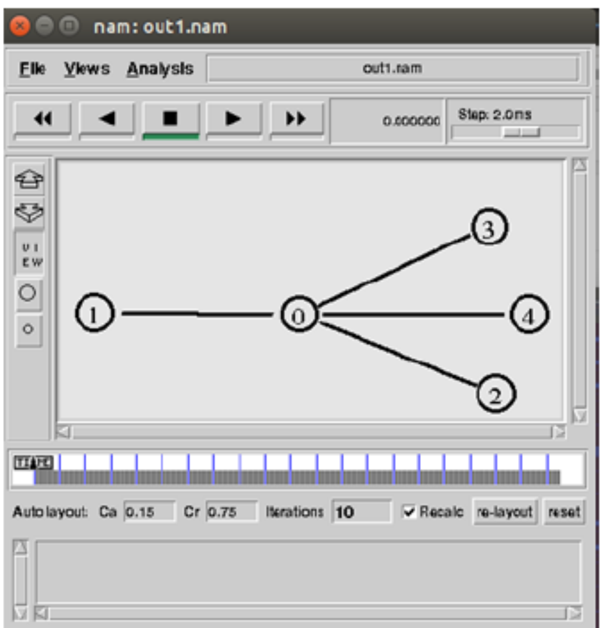

Fig. (3). Network topology used in the simulation.

The most critical area of the RED queue is the area between the minimum and maximum thresholds; because RED is always trying to stabilize the average queue size in this area [23, 24]. For this reason, we chose the midpoint between the minimum and maximum thresholds to study the impact of modifying the drop probability on the performance of RED. The average queue size at this point is an indicator of accumulative-aggressive traffic. Hence, increasing the drop probability to *0.5* will reduce this aggressive traffic and smoothen it before it reaches the most aggressive drop probability.

Fig. (**6**) shows that HRED experiences less drop oscillations comparing to the other strategies. The drop peak of HRED was at time *6s* then the drop rate went back to normal level at time *7s*. It is clear from Fig. (**4**) that the throughput of HRED was not affected by that drop, because the sources had received drop notifications smoothly and gradually. On the other hand, RED's drops had been quick and rapid from time *0s* to *2s*, as illustrated in Fig. (**5**), which caused early and quick drop notifications to the network sources; resulting in dramatic reduction in sending rate which reduced the throughput of RED at Fig. (**4**).

This scenario is one of the few scenarios that show better performance of TD over RED. Scenarios like this brought the idea of RED adaptation technique. In this research, we took Gentle-RED as an example of the RED adaptation strategies. The use of the *Gentle* parameter of RED smoothen the drop rate as illustrated in Fig. (**4**). The throughput is also increased to higher level comparing to the original RED. Tail Drop (TD) throughput suffers from ups and downs due to the full queue problem which is unavoidable behavior of TD. However, the throughput of TD is still higher than the Gentle-RED for this scenario; as illustrated in Fig. (**4**).

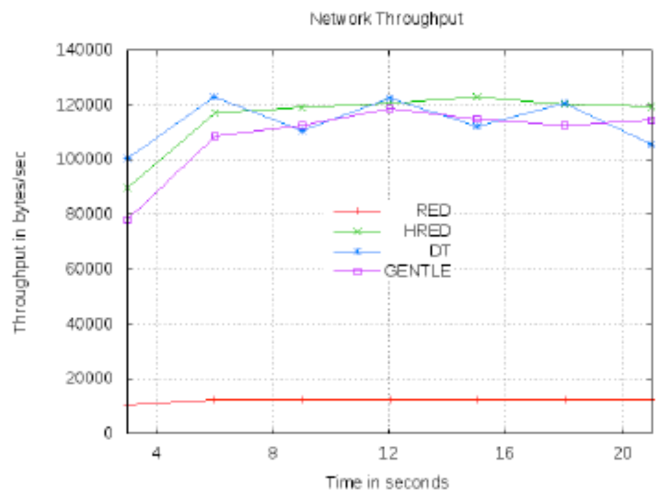

Fig. (4). Network throughput.

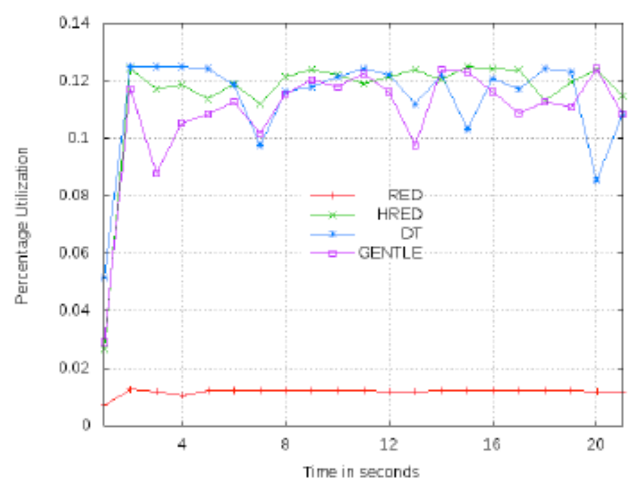

Fig. (5). Link Utilization.

The link utilization of TD in Fig. (**5**) is also higher than the link utilization of Gentle-RED. It is clear that the adaptation of the *Gentle* parameter of RED has increased the performance of RED in a more tuned manner than TD but with the cost of less throughput and link utilization. Our strategy HRED adapted RED at the midpoint of threshold which decreased the oscillations of the drop rate in Fig. (**6**) with less reduction of the source sending rates. Therefore, HRED has balanced the packet arriving and queue drain processes; resulting in higher throughput as shown in Fig. (**4**). The link utilization is also higher for HRED in Fig. (**5**).

When and how to react to traffic bursts are very important questions that would affect the overall performance of RED. For example, a quick and aggressive reaction to short bursty traffic by RED as in Fig. (**6**) had reduced the overall performance. The dropping peak of RED was at time *3s* which is too early compared to the other strategies in the study. This caused early and successive congestion signals that were triggered to the TCP sources; causing high reduction in sending rates which is the main reason of the poor performance of RED for the rest of the simulation time.

Adaptation techniques; such as Gentle-RED are better than RED in smoothen out traffic in such a scenario[1]. It is clear in Fig. (**6**) that the Gentle-RED had been reacting smoothly with the bursts of traffics by increasing the drop rates gradually until time *6s* when the congestion became a serious problem. Even after reaching the peak, the drop rate of Gentle-RED continued in higher level than RED. This is not an indicator of severe congestion; rather it is a functionality of the RED adaptation techniques to distribute the congestion signals in a manner that avoids the dramatic reduction in sending rates. This is clear in Fig. (**5**) when the Gentle-RED's link utilization continued with high rates while RED failed to recover from congestion.

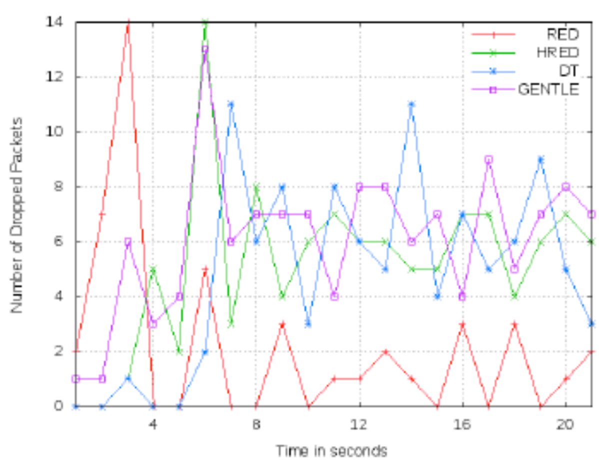

Fig. (6). Packet drop rate.

HRED provides an adaptation technique to absorb bursty traffic. The drop rate of HRED is less oscillated than the other strategies in the study. Fig. (**6**) shows a more uniformed figure for the drop rate of HRED depicted as higher link utilization in Fig. (**5**) and higher throughput in Fig. (**4**). Also, HRED outperformed the TD strategy as illustrated in Figs. (**5** and **6**).

CONCLUSION

In this article, we proposed an enhancement to the drop probability calculation of the Random Early Detection (RED) strategy for congestion control. This enhancement over RED is conducted to increase the performance of TCP networks. We have studied the drop rate when the average queue size is at the midpoint between the minimum and maximum thresholds. The proposal suggests a nonlinear adjustment for the drop rate in this area. Hence, we call this strategy as the Half-Way RED (HRED). Our strategy is tested using the NS2 simulator and compared with some queue management strategies including RED, TD and Gentle-RED. The calculated parameters are: throughput, link utilization and packet drop rate. Each performance parameter has been plotted in a separate figure; then the robustness of each strategy has been evaluated against these parameters. The results suggest that this function has enhanced the performance of RED-like strategies in controlling congestion. HRED has outperformed the strategies included in this article in terms of throughput, link utilization and packet loss rate.

CURRENT & FUTURE DEVELOPMENTS

We are currently working on the adaptation strategies of RED variants such as ARED and Gentle-RED. In the future, we are planning to compare the performance of these strategies with CoDel [15]. We are also interested in creating simulations with more complex topologies to investigate how these approaches would handle wireless links or asymmetric broadband links. Moreover, we are planning to compare HRED with some of our existing RED's optimizations [25].

CONSENT FOR PUBLICATION

Not applicable.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] G. Elias, B. Gafni, T. B. Shachar , R. Kriss and E. Rubinstein , "Flexible Allocation of Packet Buffers". USA Patent US20170201469A1, 13 July 2017.

[2] C. M. Ramaiah and P. R. Dhanabalan, "System for bandwidth optimization with initial congestion window determination". USA Patent 9929956, 2018.

[3] N. Nigar and M. A. Azim, "Fairness Comparison of TCP Variants over Proactive and Reactive Routing Protocol in MANET," *International Journal of Electrical and Computer Engineering,* vol. 8, no. 4, p. 8, 2018.

[4] S. K. Bisoy, B. Pati, C. R. Panigraph and P. K. Pattnaik, "Analysis of TCP Variant Protocol Using Active Queue Management Techniques in Wired-Cum-Wireless Networks," in *Computational*

[1]This is not always the case. The adaptation techniques also have their problems under some scenarios. A research paper by the authors titled "Revisiting the Gentle Parameter of The Random Early Detection (RED) for TCP Congestion Control" is highlighting these problems. To appear in the journal of communications, Vol. 14, No. 3, 2019.

Intelligence in Data Mining, part of: Advances in Intelligent Systems and Computing, Singapore, 2017.

[5] C. A. Grazia, N. Patriciello and M. Klapez, "A cross-comparison between TCP and AQM algorithms: Which is the best couple for congestion control?," in *14th International Conference on Telecommunications (ConTEL)*, Zagreb, Croatia, 2017.

[6] K. Nichols, V. Jacobson, A. M. Ed and J. . I. Ed, "Controlled Delay Active Queue Management," IETF, 2018.

[7] P. K. Dash, S. K. Bisoy, N. K. Kamila and M. Panda, "Parameter Setting and Stability of PI Controller for AQM Router," in *Handbook of Research on Wireless Sensor Network Trends, Technologies, and Applications*, 2017, pp. 371-393.

[8] N. Sharma, S. S. Rajput, A. K. Dwivedi and M. Shrimali , "P-RED: Probability Based Random Early Detection Algorithm for Queue Management in MANET," in *Advances in Computer and Computational Sciences. Advances in Intelligent Systems and Computing*, Singapore, 2018.

[9] P. Szilagyi and C. Vulkan , "Network-side buffer management". USA Patent US9961585B2, 1 5 2018.

[10] M. Casoni, C. Grazia, M. Klapez and N. Patriciello, "How to avoid TCP congestion without dropping packets: An effective AQM called PINK," *Computer Communications,* vol. 1, pp. 49-60, 2017.

[11] K. Nichols and V. Jacobson, "Controlling Queue Delay"," *ACM Magazine,* vol. 10, no. 5, p. 20, 2012.

[12] G. Fairhurst and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)," IETF RFC-8087, 2017.

[13] T. Burbridge, A. Smith and P. L. EARDLEY , "Processing data items in a communications network". USA Patent US20180091431A1, 2018.

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

[15] K. Nicholas, V. Jacobson, A. McGregor and J. Lyengar, "Controlled Delay Active Queue Management'," IETF, draft-ietf-aqm-codel-04, 2016.

[16] K. S. Bisoy , K. P. Pattnaik, B. Pati and P. R. Chhabi, "Design and analysis of a stable AQM controller for network congestion control," *International Journal of Communication Networks and Distributed Systems,* vol. 20, no. 2, 2018.

[17] S. Mohapatra, S. Bisoy and P. Dash, "Stability Analysis of Active Queue Management Techniques," in *International Conference of Man and Machine Interfacing (MAMI)*, 2015.

[18] C. W. Feng, L. F. Huang, C. Xu and Y. C. Chang, "Congestion Control Scheme Performance Analysis Based on Nonlinear RED," *IEEE Systems Journal,* vol. 4, no. 11, pp. 2247 - 2254, 2015.

[19] Karmeshu, S. Patel and S. Bhatnagar, "Adaptive Mean Queue Size and its Rate of Change: Queue Management with Random Dropping," *Telecommunication Systems,* vol. 65, no. 2, June 2017.

[20] X. Jiang, G. Jin and J. Yang, "LRURC: A Low Complexity and Approximate Fair Active Queue Management Algorithm for Choking Non-Adaptive Flows," *IEEE Journals & Magazines,* vol. 15, no. 4, pp. 545-548, 2015.

[21] L. Sup, R. de Moraes and A. Bauchspiess, "Explicit Non-Congestion Notification: A New," in *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Valencia, Spain, 2017.

[22] D. R. Augustyn, A. Domański and J. Domańska, "A Choice of Optimal Packet Dropping Function for Active Queue Management," in *International Conference on Computer Networks*, 2010.

[23] P. Wang, D. Zhu and X. Lu, "Active queue management algorithm based on data-driven predictive control," *Telecommunication systems,* vol. 64, no. 1, p. 103–111, January 2017.

[24] S. K. Bisoy and P. K. Pattnaik, "Design of feedback controller for TCP/AQM networks," *International Journal o Engineering Science and Technology,* vol. 20, no. 1, pp. 116-132, February 2017.

[25] 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.