International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 05 | May-2018 www.irjet.net p-ISSN: 2395-0072

ModRED : Modified RED an Efficient Congestion Control Algorithm for Wireless Network

Krishna Kachhad[1], Dr. Amit Lathigara[2]

[1]Student,Department of Computer Engineering, Marwadi Education Foundation – Faculty of PG Studies & Res. in Eng. & Tec. Rajkot, Gujarat, India
Associate Professor, Department of Computer Engineering, Marwadi Education Foundation – Faculty of PG Studies & Res. in Eng. & Tec. Rajkot, Gujarat, India

---

Abstract – *Now a day's the number of applications running over computer networks has been increasing tremendously, which increased the number of packets running over the network, which ultimately results in congestion. Random Early Detection (RED) is the most well-known and widely used method to avoid congestion. To overcome the limitations of the basic RED algorithm, researchers proposed several variants of RED. Our aim is to design an efficient congestion control algorithm to handle equally for all types of incoming traffic. This paper presents a minimal adjustment of RED called Modified RED (ModRED), in which the packet dropping probability is divided into three equal sections with using AIMD (Additive Increase Multiplicative Decrease) algorithm. The NS-2.35 simulation results show that our new ModRED algorithm gives better performance than RED. Comparisons are done in terms of throughput, good put, PDR and delay. Compared with RED, ModRED increased throughput, good put, PDR by 10.58%, 10.48%, 5.72% and decreased delay by 5.34%*

Key Words: Congestion Control, Random Early Detection (RED), Modified RED (ModRED), Active Queue Management (AQM), Transmission Control Protocol (TCP), TCP New Reno, Throughput, Good put, Packet Delivery Ratio (PDR), Delay

1. INTRODUCTION

A wireless network is a flexible data communications system, which uses wireless media such as radio frequency technology to transmit and receive data over the air. The Transmission Control Protocol (TCP) is a transport layer protocol that provides reliable data transfer, connection oriented service, flow control and congestion control. TCP is mostly used protocol in current communication network and over the internet. TCP supports mechanisms such as slow start, congestion avoidance, fast retransmission and fast recovery. Nowadays, the rapid growing of network technology and applications, network congestion has been increasingly serious and the congestion control is becoming more and more urgent. Network congestion is an essential factor affecting quality of service of the network. Congestion occurs when total demand for a resource exceeds the capacity of the resource. Congestion control strategy based on intermediate node (router) have been proposed and gotten attention to compensate for the lack of the source end TCP. Results of congestion include a high delay, decrease throughput and even global synchronization. The aim then would be to control congestion or more ideally, avoid congestion. Congestion control techniques have been developed called Active Queue Management (AQM). A congestion control scheme based on active queue management has become a research hot spot in the industry.

The most basic AQM approach is the Drop Tail (DT) scheme where packets arriving in a queue are dropped with probability one when the queue is full. However, Drop Tail can cause that all TCP flows through the congested queue reduce their transmission rates at the same time, because each TCP source reduces its window every time it detects a packet loss. Moreover, in each congestion episode, there is a high chance of dropping a packet from each active flow. This phenomenon is known as global synchronization. Drop Tail queue suffer the problem of global synchronization in which queue is over utilized and underutilized at alternative period of time. In 1993, Floyd and Jacobson [1] proposed an algorithm called Random Early Detection (RED). RED queue mechanism is simple and effective to overcome global synchronization. Some modification on RED algorithm have been proposed such as CRED [2], FLRED [3], TRED [4], ELALRED [5], ASRED [6], Fuzzy-CARED [7], Gentle-BLUE [8] and URED [9]. RED queue algorithm calculates the average queue size (avg). The average queue size is compared with two thresholds: a minimum threshold ($Minth$) and a maximum threshold ($Maxth$). Briefly, the algorithm works by maintaining an average queue size. As the average queue size varies between the minimum and maximum thresholds, the packet dropping probability linearly changes between zero and maximum drop probability $Pmax$.

RED uses a mechanism early detection of packet drop without waiting to queue overflow. When congestion will happen, router discards the arriving packets with certain probability. This can inform the sender to adjust size of sending window before congestion happen. RED gets the average queue length of router to predict the network congestion. RED algorithm maintains two parameters maximum threshold and minimum threshold. RED uses the weighted average function to predict the average queue length and compares the results with the minimum and maximum threshold predefined. If it is less than the minimum threshold, the arrival packet will be forwarded normally. If estimated result is greater than the maximum threshold, all the arriving groups will be discarded. If the average queue length increases above $Minth$ but is below $Maxth$, RED drops incoming packets with a probability proportional to the average queue length linearly. The RED queue algorithm works on three conditions. RED is an improvement over simple Drop Tail. RED avoids the TCP starvation problem and global synchronization but compared with Drop Tail, RED exhibits a lower delay and a higher throughput and packet loss.

RED mechanism used same dropping probability for all type of traffic; if average queue size between the minth and maxth dropping probabilities are same for all type of traffic. To overcome the limitation of packet dropping probability we propose new algorithm using existing RED algorithm called Modified RED (ModRED). This paper presents a congestion control scheme Modified RED with its packet dropping probability function divided into three sections. ModRED scheme used three dropping probabilities based on different type of incoming traffic. For different dropping probabilities ModRED algorithm used the AIMD (Additive Increase Multiplicative Decrease) algorithm. Basically, this algorithm is used for TCP congestion control. Simulation is done in NS-2.35 simulator. Simulation results are compared with RED with our proposed ModRED algorithm. Results are in terms of throughput, good put, Packet Delivery Ratio (PDR) and delay.

The rest of this paper is organized as follows: section 2 presents the literature review of some papers while section 3 presents our proposed ModRED algorithm. Section 4 presents and discusses simulation results before the paper concludes with future work in section 5.

2. RELATED WORKS

In wireless network, congestion is major issues there are different queue management techniques used to reduce it. This section discusses most widely known congestion control methods related to this work. Comparisons of various queue mechanisms are described in Table-1.

CRED (Hemi-Rise cloud model) [2] was proposed nonlinear packet loss strategy was used and uncertainty of parameters were improved. CRED algorithm combines Drop Tail and RED algorithm and remove the randomness and fuzziness of RED algorithm. CRED algorithm effectively controls oscillation of the average queue length. As a result, network congestion was controlled and network resource was used effectively.

FLRED (Fuzzy Logic Random Early Detection) [3] proposed a new method, which extends RED by fuzzy logic to overcome linearity and parameterization problems. FLRED method uses two congestion indicators that is aql and $Dspec$ to avoid congestion at early stage. A discrete time queue model was used to improve FLRED.

TRED [4] presents a minimal adjustment of RED called Three Section Random Early Detection based on nonlinear RED. In TRED packet dropping probability is divided into three sections that are light, moderate and high loads. TRED aimed at solving RED's link underutilization and large delay and high traffic load problems. TRED enhances the ability to regulate congestion, improving resource utilization and scheme's stability.

ELALRED (Efficient Learning Automata Like RED) [5] proposed a another method that avoid the congestion for wired networks. ELALRED is founded on the RED principle, with LAL philosophy and aims to optimize the value of the average size of queue and reduce the total loss of packets. ASRED (Adaptive Sigmoid RED) [6] was proposed novel adaptive queue management intelligent algorithm. In this paper new active queue management algorithm adopts a new formula to calculate the discard packet rate. This discard packet rate can be calculated according to the changes of average queue.

Fuzzy–CARED (Fuzzy Cautious Adaptive RED) [7] proposed fuzzy logic which uses set of rules to predict the congestion in the network. CARED is designed for heterogeneous network. Fuzzy-CARED is designed to improve CARED in which $maxp$ increases and decreases after detecting congestion in network.

Gentle-BLUE [8] was proposed a new method for active queue management. Gentle-BLUE extends the BLUE method by providing dynamic mechanism for calculating the dropping probability based on status of the queue length. The dropping value is dynamically calculated based on queue length and remaining buffer capacity.

URED (Upper Threshold RED) [9] proposed an efficient congestion control algorithm to design for denial of service attacks. In this research propose an algorithm, with minimal changes to RED which provides solutions to avoid congestion of network services by new threshold Uth. In URED uses three threshold that are minimum threshold, maximum threshold and upper threshold and it will also increase adaptability of RED.

ENCN (Explicit Non-Congestion Notification) [10] was proposed new AQM approach for TCP networks. ENCN presents a new work which unlike ECN, instead of notifying congestion in the router, it notifies non-congestion on the path. To overcome unwanted empty queue phenomenon and to take advantage of no congestion state of the queue, propose an AQM technique called ENCN.

ECC (Early Congestion Control) [11] proposed another new approach to improve the performance of TCP in ad hoc networks. In this paper, ECC a new cross layer mechanism that dynamically changes the value of the window field from TCP headers according to the utilization of the router queue. This new method is TCP friendly and does not require TCP changes.

Table -1: Comparisons of Various Queue Mechanisms

Queue Name	Dropp-ing Proba-bility	TCP Variants	Queue Size	ECN Notifi-cation	Comp-lexity
CRED[2]	Yes	TCP	300 Packet	Yes	Yes
FLRED[3]	Yes	TCP	20 Packet	No	Yes
TRED[4]	Yes	TCP NewReno	120 Packet	No	Yes
ELALRED [5]	Yes	TCP	1000 Packet	No	Yes
ASRED[6]	Yes	TCP	300 Packet	No	No
Fuzzy-CARED[7]	Yes	TCP/UDP	1282 Packet	No	No
Gentle-BLUE[8]	Yes	TCP	20 Packet	No	Yes
URED[9]	Yes	TCP	100 Packet	No	No
ENCN[10]	No	TCP NewReno & TCP Sack	17 Packet	Yes	No
ECC[11]	No	TCP NewReno	97 Packet	No	No

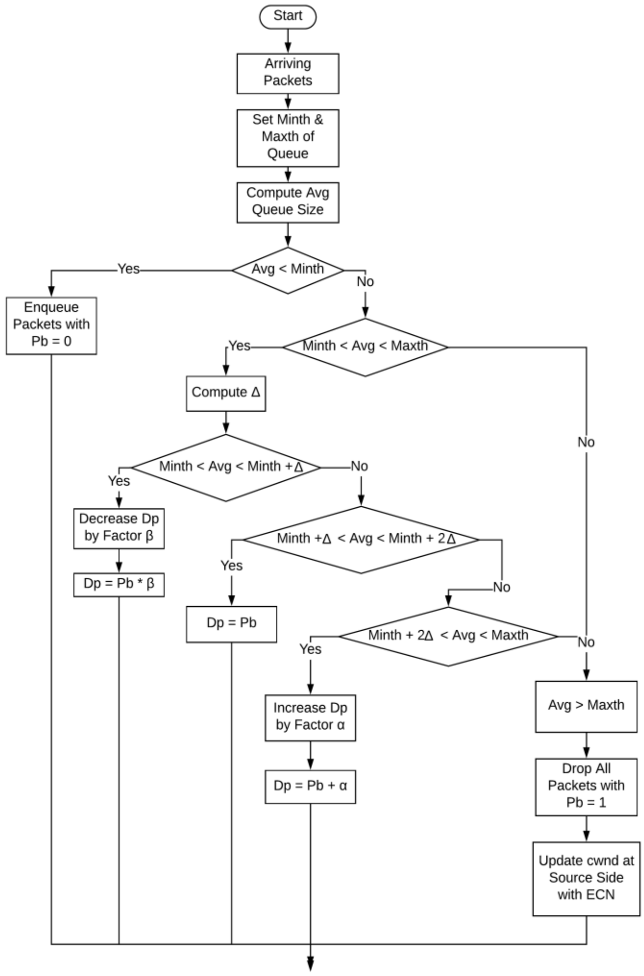

Fig -1: Flowchart of the ModRED algorithm for queue

3. PROPOSED MODRED ALGORITHM

In ModRED algorithm we have used three dynamic dropping probabilities based on incoming traffic and it is efficient to use different traffic patterns. ModRED algorithm is founded on the RED principle, with AIMD algorithm and aims to different dropping probability for different traffic load. As in RED and other enhanced RED algorithm only use one dropping probability and treat same for all type of traffic. In ModRED scheme use three dropping probability based on incoming traffic. Based on which type of load the dropping probabilities of packets are calculated. The dynamic dropping probability is calculating base on AIMD (Additive Increase Multiplicative Decrease) algorithm. Basically AIMD algorithm is used for TCP congestion control. Proposed algorithm also maintains congestion at receiver side.

In proposed algorithm two threshold minimum threshold ($Minth$) and maximum threshold ($Maxth$) are used. The average queue length of the queue is calculated same as the core of RED algorithm.

$$Avg = (1 - c) * avgp + c * q \qquad (1)$$

Where Avg is the average of the queue length, c is the constant (c = 0.02), $avgp$ is the previous average of the queue length (first time $avgp = 0$) and q is the total queue length. Now using the average formula compute the average of the queue length then average queue size compares with the minimum threshold and maximum threshold.

The flowchart of the ModRED algorithm for queue in the router is illustrated in Fig–1. If average queue length is less than $Minth$, ($Avg < Minth$), then enqueue the packets into the queue because the queue is empty. If average queue length between the $Minth$ and $Maxth$ ($Minth < Avg < Maxth$), then calculates the dropping probabilities of packets. In this condition calculate the different dropping probabilities based on divided into three equal portions. And if the average queue length is greater than $Maxth$, ($Avg >$

Maxth), all incoming packets are dropped and update congestion window at source side with ECN (Explicit Congestion Notification). The aim of ModRED algorithm works on main three divided equal portions for dropping probabilities. If *Avg* between the *Minth* and *Maxth* first calculate the delta Δ = (*Maxth* – *Minth*/2). Calculating dropping probability (Dp) based on divided into three equal sections.

- If $Minth < Avg < Minth + \Delta$, then decrease *Dp* by factor β
  $$Dp = Pb * \beta \quad (2)$$

Where $Pb = maxp\ (Avg - Minth\ /\ Maxth - Minth)$, $\beta = 0.09$. This condition near to the *Minth* means queue is empty. Therefore need to decrease the dropping probability.

- If $Minth + \Delta < Avg < Minth + 2\Delta$, then calculate the dropping probability as *Pb* equation
  $$Dp = Pb \quad (3)$$

- If $Minth + 2\Delta < Avg < Maxth$, then increase *Dp* by factor α
  $$Dp = Pb + \alpha \quad (4)$$

Where $Pb = maxp\ (Avg - Minth\ /\ Maxth - Minth)$ and $\alpha = 0.25 * cur\_max\_p$. This condition near to the *Maxth* means queue is not empty. Therefore need to increase the dropping probability for reduce the congestion.

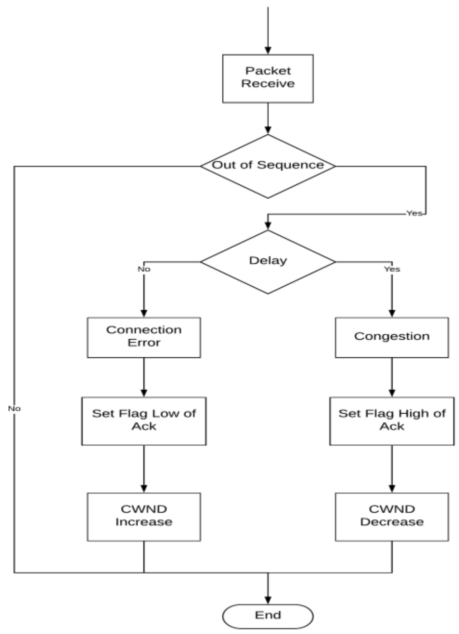

Fig -2: Flowchart of the ModRED algorithm with TCP NewReno for receiver end

The flowchart of the ModRED algorithm with TCP NewReno for receiver end is illustrated in Fig-2. In ModRED with TCP NewReno algorithm also effective when receiver side congestion generated. At receiver side, control the congestion by send the flag bit low (0) and high (1) of acknowledgement to sender. When packet receive with out of sequence or bit pattern is different, due to this generate the delay. If delay is generating it means congestion occurs and delay is not generating it means some connection error occurs. Based on congestion generated, the flag bit is set to high (1) and decrease the congestion window at sender side. If connection error generated, the flag bit is set to low (0) and increase the congestion window at sender side.

In summary, the proposed ModRED algorithm calculates the dynamic dropping probability based on incoming traffic and also handle the congestion at receiver side congestion occurs. The equations of ModRED's packet dropping probability are shown in (2), (3) and (4).

4. SIMULATION RESULT

The software NS–2.35 is adopted for network simulation. Following table gives the simulation parameters in details. Graphs are generated using Gnuplot.

Table -2: Simulation Parameters

Parameters	Value
No. of Nodes	25, 50, 75, 100
Area	500 x 500 Meter
Traffic Area	TCP NewReno
Simulation Time	29.0 s
Queue Length	50
Packet Size	512 Bytes
Queue Weight	0.001
Minimum Threshold	5
Maximum Threshold	15
Protocol	AODV

The entire simulation time is set to 29 s and queue length is set to 50. Some parameters are set as described in Table-2 and the rest is used the default values in NS-2.35. ModRED algorithm was compared to the RED algorithm and TCP used NewReno with the packet size of 512 bytes. The simulations were performed with 25, 50, 75 and 100 nodes.

4.1 Throughput

Throughput is a measure of receiving packets per second at the network receiver end. Fig-3 shows the throughput comparisons of RED and ModRED for different nodes. Throughput is measured in kbps (kilobytes per second). Simulation results show that the throughput of ModRED is 10.58% increase than RED algorithm.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 05 | May-2018 www.irjet.net p-ISSN: 2395-0072

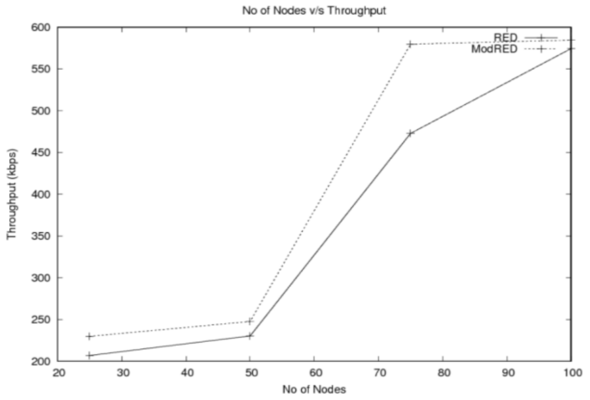

Fig -3: No. of nodes v/s Throughput (kbps)

4.2 Good put

Good put is the average rate of successful packet over a network. Good put is measured in kbps. Fig-4 shows the good put comparisons of RED and ModRED for different nodes. Simulation results show that the goodput of ModRED is 10.48% increase than RED algorithm.

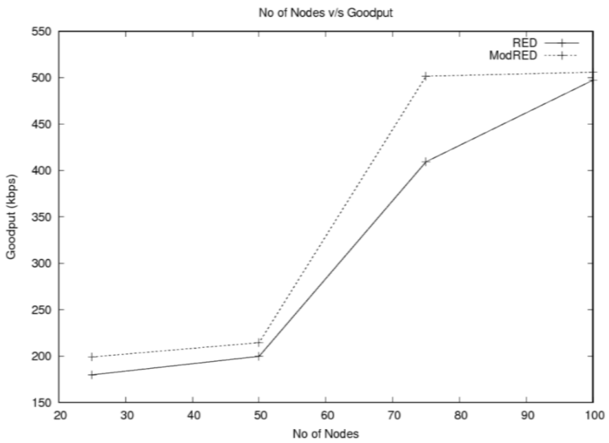

Fig -4: No. of nodes v/s Goodput (kbps)

4.3 Packet Delivery Ratio (PDR)

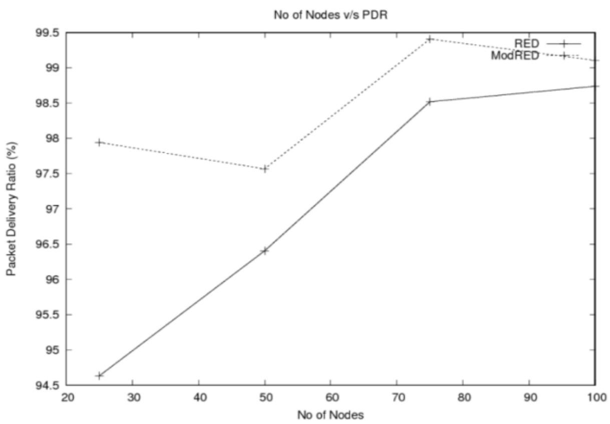

Fig -5: No. of nodes v/s PDR (%)

Packet Delivery Ratio (PDR) is a measure of total successful data packets received by the destinations to those generated by the sources. PDR is measured in %. Fig-5 shows the PDR of RED and ModRED for different nodes. Simulation results show that the PDR of ModRED is 5.72% increase than RED algorithm. ModRED algorithm controls the congestion because of the dynamic dropping probability and therefore packets are delivered more compared to the RED algorithm.

4.4 Delay

Delay is the difference between stop time and start time. Delay is measured in ms (millisecond). Fig-6 shows the delay of RED and ModRED for different nodes. Simulation results show that the delay of ModRED is 5.34% lower than RED algorithm.

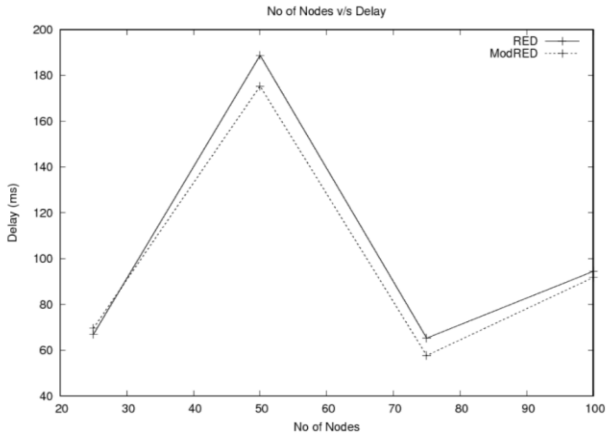

Fig -6: No. of nodes v/s Delay (ms)

Simulation results show that performance of ModRED is better than RED algorithm. ModRED algorithm gives higher throughput, better goodput, higher PDR and lower delay as compared to the RED algorithm.

5. CONCLUSIONS

Congestion is one of the key issues related to network performance. RED is one of the AQM methods that have been developed mainly to control congestion at an early stage. However, RED algorithm doesn't have dynamic dropping probability base on incoming traffic. This paper proposed a new congestion control method called Modified RED (ModRED). ModRED algorithm is using AIMD (Additive Increase Multiplicative Decrease) algorithm for dynamic dropping probability. By introducing new dynamic dropping probability, the performance of ModRED is improved. Simulation results show that ModRED is more efficient than RED. The new algorithm increases throughput, goodput, PDR and decreases delay than RED algorithm.

FUTURE WORK

For future work, we are interested in studying on ModRED with different TCP variants like TCP vegas, TCP sack, TCP Westwood etc because lots of research on RED with different TCP variants.

REFERENCES

[1] Floyd, Sally, and Van Jacobson. "Random early detection gateways for congestion avoidance", *IEEE/ACM Transactions on Networking (ToN)*, vol. 1(4), pp. 397-413, 1993.

[2] Zhao Yuhong, MA Zhonggui, Zheng Xuefeng, Tu Xuyan "An Improved Algorithm of Nonlinear RED Based on Membership Cloud Theory", *Chinese Journal of Electronics*, vol. 26(3), pp. 537-543, 2017.

[3] Abualhaj, Mosleh M., Ahmad Adel Abu-Shareha, and Mayy M. Al-Tahrawi. "FLRED: an efficient fuzzy logic based network congestion control method", *Springer Neural Computing and Applications*, pp. 1-11, 2016.

[4] Feng, Chen-Wei, et al. "Congestion control scheme performance analysis based on nonlinear RED", *IEEE Systems Journal*, vol. 11(4), pp. 2247-2254, 2015.

[5] Mahajan, Shalu G. "Efficient LALRED for Congestion Avoidance Using Automata-Like Solution", *International Conference on Emerging Information Technology and Engineering Solutions (EITES),* IEEE, Trivandrum, pp. 1-6, 2015.

[6] Zhang, De-Gan, et al. "Novel Adaptive Queue Intelligent Management Algorithm", *12th International Conference on Mobile Ad Hoc and Sensor Systems (MASS),* IEEE, Taiwan, pp. 483-485, 2015.

[7] Ravindra, Shah Ravija, and Ashwini Patil. "Fuzzy Cautious Adaptive Random Early Detection for Heterogeneous Network", *Fifth International Conference on Advances in Computing and Communications (ICACC),* IEEE, Kochi, India, pp. 65-68, 2015.

[8] Adeeb Saaidah, Mohd Zalisham, Mohd Fadzli and Hussein Abdel-jaber, "Gentle-BLUE: A New Method for Active Queue Management", *3rd International Conference on Advanced Computer Science Applications and Technologies (ACSAT),* IEEE, Amman, Jordan, pp. 67-72, 2014.

[9] Chandni M Patel, "URED: Upper Threshold RED an Efficient Congestion Control Algorithm", Fourth International Conference on *Computing, Communications and Networking Technologies (ICCCNT),* IEEE, Tiruchengode, India pp. 1-5, 2013.

[10] Sup, Luciano Mauro Arley, Renato Mariz de Moraes, and Adolfo Bauchspiess. "Explicit non-congestion notification: A new AQM approach for TCP networks", *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, Valencia, Spain, pp. 1239-1244, 2017.

[11] Talau, Marcos, et al. "Early congestion control: A new approach to improve the performance of TCP in ad hoc networks", *7th International Conference on the Network of the Future (NOF),* IEEE, Tunisia, pp. 1-6, 2016.