IJCTA, 9(41), 2016, pp. 157-164 ISSN: 0974-5572

The Comparative Analysis of RED, GF-RED and MGF-RED for Congestion Avoidance in MANETs

Makul Mahajan[1] and Mritunjay Kumar Rai[2]*

ABSTRACT

Every device in mobile ad hoc network has some buffer space required to store the packets if receiver is not able to capture it. Buffer space is usually maintained as a queue at both senders as well as receiver side. It is very important to manage the queue effectively. Various queue management techniques are used for managing the queue. This paper firstly addresses the RED active queue management technique used in most of the networks for managing the queue efficiently. Secondly it proposed a Gaussian (G-RED) and modified Gaussian function based RED (MG-RED) algorithm. Simulations are carried out using NS-2.33 simulator and it has been observed from the results that the Gaussian and modified Gaussian function based RED algorithm outperforms the traditional RED algorithm in terms of throughput, packet delivery fraction and number of packets dropped.

Keywords: RED, MG-RED, G-RED.

1. INTRODUCTION

A Mobile Ad Hoc network is a collection of mobile nodes which are connected via wireless links. The nodes in this kind of networks are mobile in the sense that they can move from one location to another location. Any mobile node need to transfer the data has to send it directly if the receiver node is in transmission range or with the help of some other mobile node if the receiver is not in range. The node in this type of network acts as a router for transferring the packets from source to destination. The topology of this type of network is highly dynamic due to mobility of nodes. MANETs are becoming very popular in these days due to its usage of anywhere and at anytime [1].

In MANETs, it's very important to manage the queue space required to store the packets that cannot be transferred to receiver instantly. Various queue management techniques are used to manage the buffer space. Drop tail technique is one of the techniques used for managing the queue in which packets are allowed to enter into the queue until the queue is not full and after the queue is full the drop tail start drops the packet at tail until queue becomes stable. Drop tail technique is inactive queue management practice which does not use any predefined information of congestion occurring in the network [1] [2].

Another technique that can be used for managing the queue in network is active queue management method. Random early detection is one of the techniques used for managing the queue actively. In RED algorithm packets start arriving in the queue until the queue does not reaches the minimum threshold. If the queue length reaches minimum threshold but does not reaches the maximum threshold, the algorithm begin dropping the packets by means of linear task of queue size. If the queue size reaches the maximum threshold, each arrived packet is dropped until queue size becomes under controlled. The advantage of using RED algorithm over drop tail technique is that the sender will be notified if congestion occurs. No

[1] Assistant Professor, *Email: makul.14575@lpu.co.in*

[2] Associate Professor, , Lovely Professional University, Punjab, *Email: raimritunjay@gmail.com*

* Corresponding Author

doubt the RED algorithm provides a very good solution for reducing the packet drop by notifying the sender for congestion. But RED uses linear function for falling the packets as the queue size increases minimum threshold. This paper investigates two new techniques that use Gaussian and modified Gaussian function to drop the packet rather than linear function. The idea is to control the number of packets dropped so that the performance of the network can be enhanced. The results shows that proposed techniques will perform superior than RED in terms of throughput, packet delivery fraction ands number of packets dropped. [1][2][3][4].

2. CONVENTIONAL RED ALGORITHM

The conventional RED algorithm used for congestion avoidance in MANETs based on average queue size. The algorithm first calculates the queue size based on number of packets within the queue then decides whether to drop the packet or let the packet to enqueue. The formula used for calculating the average queue size is given by:

$$queue\_a = (1 - p) * queue\_a + s$$

Where Queue_avg is the calculated average queue size, s is the instantaneous queue size and p is the weighted factor which is usually constant and passed during simulation. After calculating the average queue size it compares the same with minimum and maximum threshold provided for the queue. If the average queues size is less than the minimum threshold then all the arriving packets will be queued as they come for transmission in the queue. If the average queue size is greater than the minimum threshold and less than the maximum threshold then it start dropping the packets as they come using the linear function for dropping. It means algorithm drops the packets using some function which is linear role of average queue size and is given by [7] [8] [9].

$$Pb = va * queue\_ax + vb$$

Where Pb is the new packet drop probability when the average queue size exceed the minimum threshold. va and vb are constant which are calculated using the following equation

Va = 1/minthresh

Vb = minthresh/(maxthresh-minthresh)

In addition to above the algorithm is responsive to following parameters

Avg buffer Length (Queue_avrg)

Min Threshold (minthresh)

Max Threshold (maxthresh)

Max Drop Probability (maxpb)

The basic algorithm for the conventional RED algorithm is:

Step 1. Estimate the average length of queue (Queue_avrg).

Step 2. If Queue_avrg<minthresh then
Queue the packet and no drop occurs.

Step 3. If Queue_avrg>minthresh and Queue_avrg<maxthresh then
Packets should be dropped with some probability calculated using linear function.

Step 4. If Queue_avrg>maxthresh then
Packets must be dropped until length of queue reaches between minthresh and maxthresh.

Step 5. Exit.

Most of the simulation techniques uses RED algorithm as congestion avoidance than the drop tail due to various reasons. One of the main reasons is that the RED algorithm uses the concept of early detection of congestion. Now if the congestion in the network is detected early then probability of queue becoming full can be avoided hence network performance can be increased. Secondly RED algorithm uses probability for calculating packet drop, which is the main factor of controlling the queue size. But on the pessimistic side RED algorithm uses linear function for calculation the probability. Hence in the times of high congestion in the network the probability of dropping the packet increases as the queue size increase. These results in more number of packets dropped hence performance degrades [8] [11].

By considering the above stated factor there is a need to work upon the function used for calculating the probability of dropping the packet when the average queue size increases the minimum threshold. In the next two sections this paper proposes the Gaussian and modified Gaussian function for calculating the probability of packet drop. The simulation consequences show that our proposed techniques work better than the conventional technique.

The figure 1 depicts the flowchart of basic task done by conventional RED algorithm. As and when packet arrives it calculates the average queue length. If average queue length is less than the minimum threshold then packets are queued. If average queue length is greater than minimum threshold but less than maximum threshold then algorithm start dropping the packets according to some probability calculated using linear function. Lastly if queue size exceeds the maximum threshold then each arriving packet is dropped until queue size reaches to certain level. The process repeats for the entire simulation time.

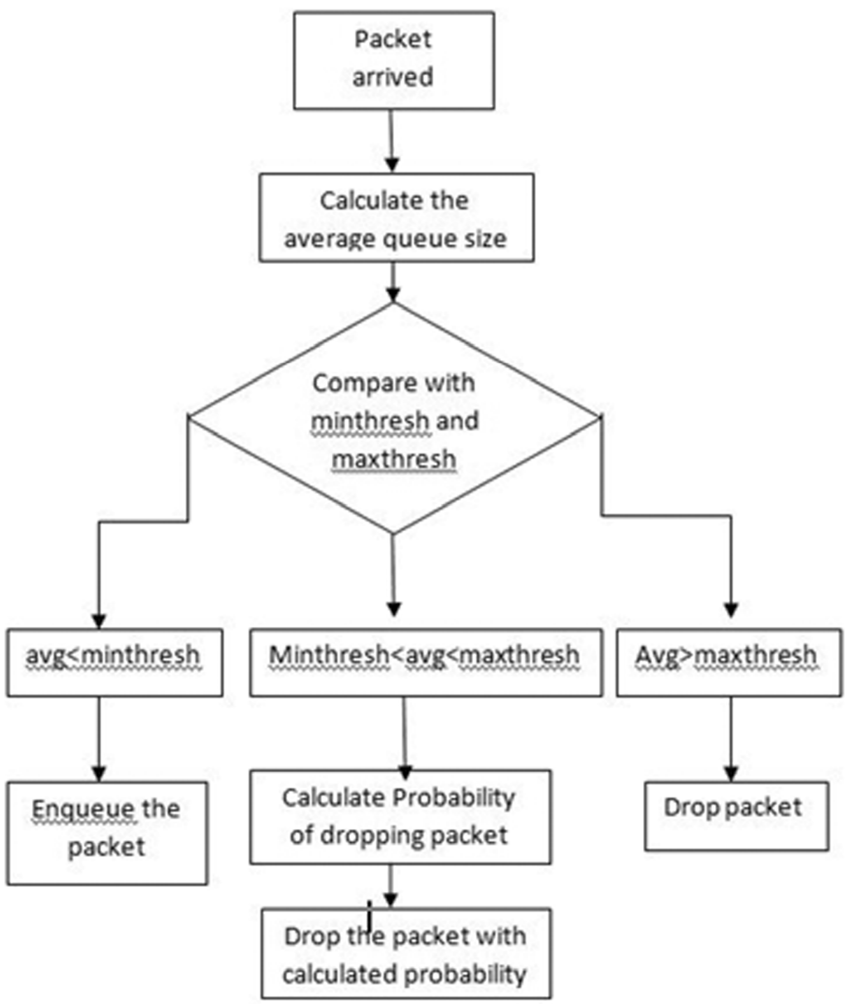

Figure 1: Basic RED algorithm working

3. PROPOSED GAUSSIAN FUNCTION RED

In this section the paper focus on the new technique for calculating the probability of packet drop in RED algorithm if the average queue size increases the minimum threshold. From the previous section it has been confirmed that there is a need to work upon the task which is calculating the drop speed. Hence after investigation the various functions it has been determined to include the Gaussian function for calculating the probability for packet drop and is given by:

Where $x = (Queue\_avrg-b)^2/2c^2$

Queue_avrg is the average queue size and is given by:

Queue_avrg = (1-f)*Queue_avrg + q*f and f is the weight of queue and is constant value given at simulation's time.

Where Pb is calculated new packet drop probability

va and vb are constant values and are as follows:

va = 1/maxthresh-minthesh

vb = minthresh/maxthresh-minthresh

$a = \cos^2(\varphi) + \sin^2(\varphi)$

$b = \sin(2\varphi) + \sin(2\varphi)$

$c = \sin^2(\varphi) + \cos^2(\varphi)$

$\varphi = v\_a*Queue\_avrg+v\_b$.

The proposed work given above will calculate the probability of dropping the packets using the Gaussian function when the average queue size exceeds the minimum threshold.

4. PROPOSED MODIFIED GAUSSIAN FUNCTION RED ALGORITHM

In the previous section we have discussed a new function for calculating the packet dropping probability. In this section we will discuss the modified Gaussian function based RED algorithm for scheming the packet drop. The normal Gaussian function uses the concept of exponential values but the modified Gaussian function introduces a new variable and the new modified Gaussian function is given by [12]:

$$pb = a * va^{-x} + vb$$

Where $x = (Queue\_avrg-b)^2/2c^2$

Queue_avrg is the average queue size and is given by:

Queue_avrg = (1-f)*Queue_avrg + q*f and f is the weight of queue and is constant value given at simulation's time.

Where Pb is calculated new packet drop probability

va and vb are constant values and are given by

va = 1/maxthresh-minthresh

vb = minthresh/maxthresh-minthresh

$a = \cos^2(\varphi) + \sin^2(\varphi)$

$b = \sin(2\varphi) + \sin(2\varphi)$

$c = \sin^2(\varphi) + \cos^2(\varphi)$

φ = v_a*Queue_avrg + v_b [12].

The proposed above algorithm is evaluated with conventional and Gaussian RED algorithm the results shows that modified Gaussian function based RED work better than Gaussian and conventional RED [12].

The algorithm for this scheme is given by:

Step 1. Estimate the average length of queue (Queue_avrg).

Step 2. If Queue_avrg<minthresh then
Enqueue the packet and no drop occurs.

Step 3. If Queue_avrg>minthresh and Queue_avrg<maxthresh then
Calculate the probability of dropping packets

Step 4. If Queue_avrg>maxthresh then
Drop the packet until length of queue reaches between minthresh and maxthresh.

Step 5. Exit.

In this algorithm the concept is to minimum the number of dropped packets so that network performance can be increased.

5. SIMULATION SETUP AND ASSUMPTIONS TAKEN

The simulation of above three algorithm discussed in the previous sections is done on NS-2.35. In our simulation the basic protocol that we taken is AODV. The following parameters need to be set up and to be assumed for the simulation:

Min Threshold: 5

Max Threshold: 15

Weight of queue: 0.001

Max dropping probability: 0.05

The algorithms have been evaluated on following parameters:

Throughput: It may be defined as total size of packets recived by all the destination nodes.

Packet Delivery Fraction: It may be defined as the ratio of number of packets received to number of packets sent by is the source in the network.

Number of dropped packets: It is the difference of number of packets sent plus forwarded and number of packets dropped. The following table shows the parameters to be taken for simulation.

Table 1

Parameter	*Value*
Routing Protocol	AODV
Area	1000*1000
Pause Time	6
Number of nodes	10,20,30,40
Simulation Time	100
Packet Size	1000b
Traffic pattern used	CBR

5.1. Outcomes and Discussion

In this section the paper discuss the various outcomes of our proposed techniques and compared with it conventional technique.

5.1.1. Throughput

The diagram 2 shows the proposed MGF-RED algorithm perform better than both GF-RED and RED. The GF-RED performs better than the RED algorithm. It has been detected from the simulation that our proposed algorithm performs better when the congestion is there in the network in term of throughput. The throughput has been calculated and is compared by varying the number of nodes in the network. In every case taken in this paper the proposed algorithm performs better.

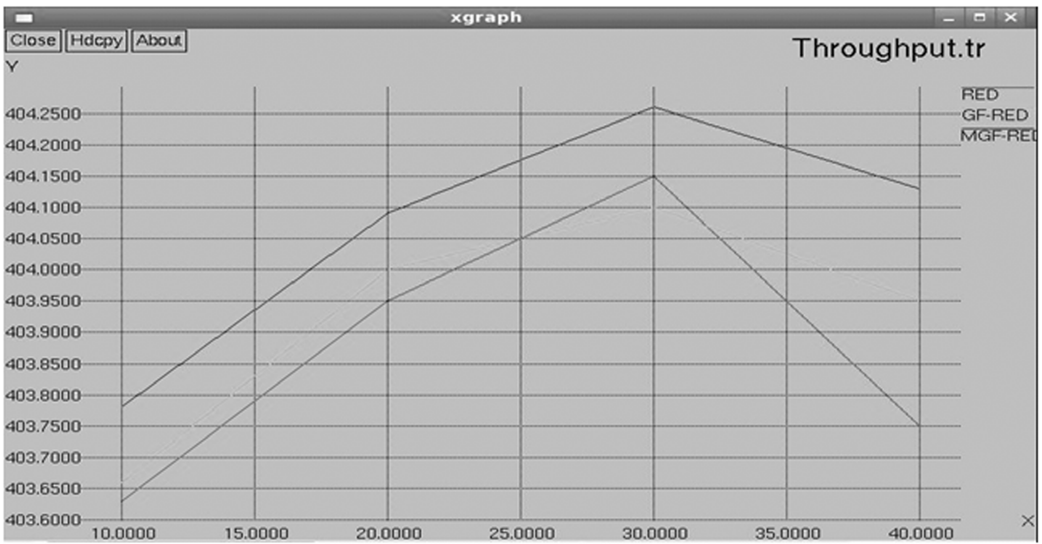

Figure 2: Throughput Versus Number of nodes

5.1.2. Packet Delivery Ratio

The diagram 3 shows the PDF of MGF-RED is greater than the both GF-RED and RED algorithm and PDF of GF-RED is greater than the RED algorithm.

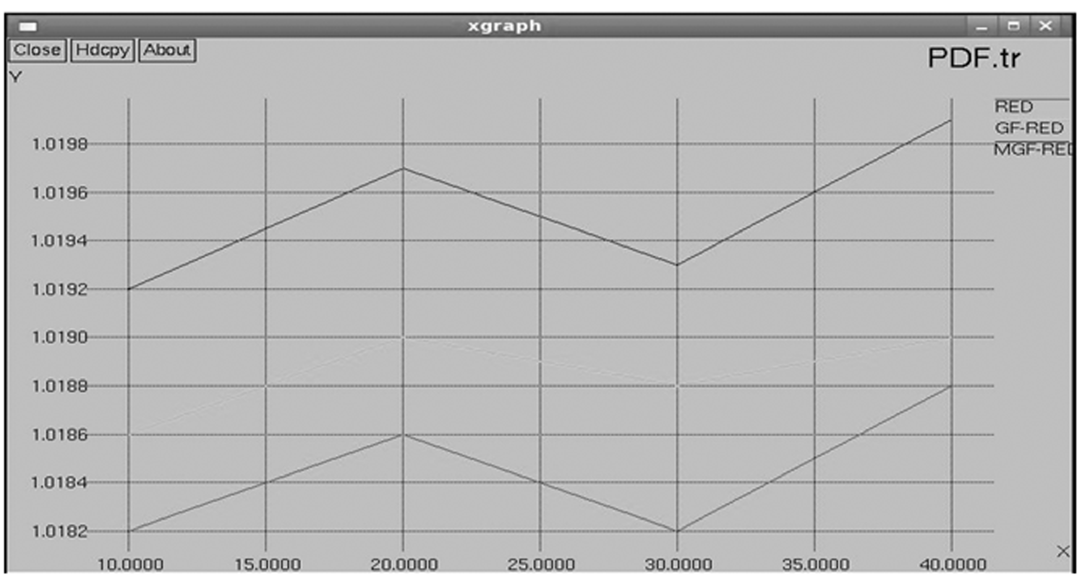

Figure 3: Packet Delivery Ratio versus Number of Nodes

5.1.3. Number of Packets Dropped

The diagram 4 shows that the MGF-RED algorithm drops fewer number of packets than the GF-RED and RED algorithm and GF-RED algorithm perform better than the RED in term of number of packets dropped.

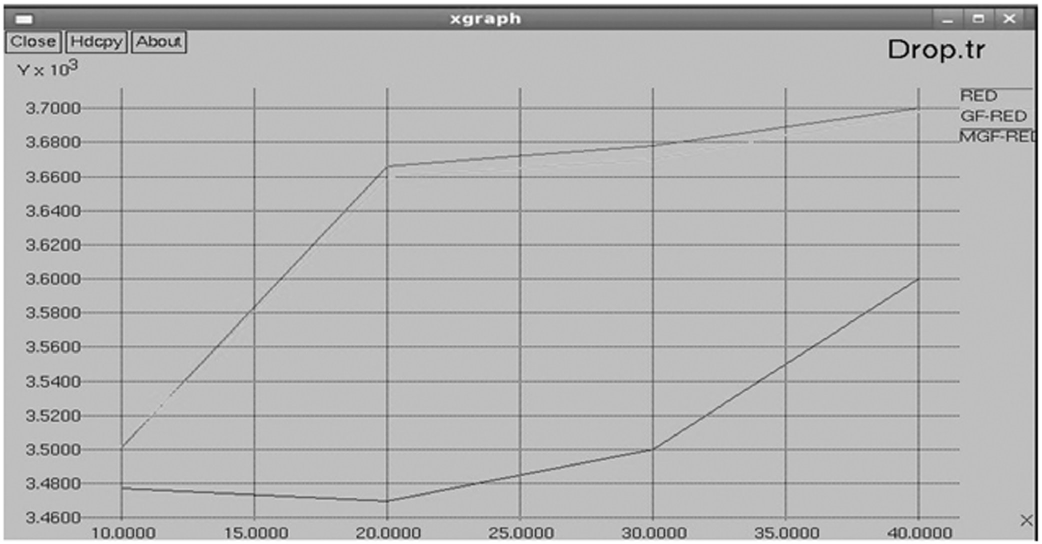

Figure 4: Number of dropped packets Versus Number of Nodes

6. CONCLUSION AND FUTURE WORK

The paper summarizes the three algorithms RED, GF-RED and MGF-RED for congestion avoidance in MANETs. It has been observed from the above discussion and outcomes that the MGF-RED algorithm gives better results than GF-RED and conventional RED in terms of throughput, PDF and number of packets dropped. GF-RED performs better than the conventional RED algorithm. This paper modifies the concept of dropping the packets in RED algorithm when average queue size exceeds the minimum threshold so that better network performance can be achieved. Simulation has been carried out by varying the network size in terms of number of node. In future the concept can be applied for other routing protocols present in MANETs and Wireless sensor networks. The performance for other protocols can be evaluated using the concept.

REFERENCE

[1] Geethu Mohandas, D. S. "Survey on Routing Protocols on Mobile Adhoc.", IEEE, pp. 514-517, 2013.

[2] S Floyd, V Jacobson, "Random early detection gateways for congestion avoidance." IEEE/ACM Trans Netw, 1(4), 397–413, doi:10.1109/ 90.251892,1993.

[3] D Lin, R Morris, "Dynamics of random early detection", in Proceedings of ACM SIGCOMM 97, Cannes, France, pp. 127–137 (1997)

[4] Chengyu Zhu, M. Ouellette and D. Y. Montuno and 0. W. W. Yang, "A comparison of active queue management algorithms using the OPNET Modeler," IEEE Communications Magazine, vol. 40, pp. 158-167, 2002.

[5] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management".

[6] M. Christiansenet al., "Tuning RED for web traffic," in Proc. ACM SIGCOMM, Stockholm, Sweden, 2000.

[7] Guan-Yi Su, C. C. "Random Early Detection Improved by Progressive Adjustment Method". Proceedings of IEEE 2008 6th National Conference on Telecommunication Technologies and IEEE, 2nd Malaysia Conference on Photonics, Putrajaya, Malaysia (pp. 250-253), 26-27 August 2008.

[8] Hussein Abdel-jaber, M. M. "Fuzzy Logic Controller of Random Early Detection based on Average Queue Length and Packet Loss Rate". SPECTS, pp. 428-432, 2008.

[9] Rahul Verma, A. I., "Active Queue Management using Adaptive RED. JOURNAL OF COMMUNICATIONS AND NETWORKS", 275-281.

[10] B. Suter, T.Y. Lakshman, D. Stiliadis, and AK Choudhury. "Buffer management schemes for supporting TCP in gigabit routers with per-flow queuing". IEEE Journal on Selected Areas in Communications, vol. 17, no. 6,1999.

[11] V. Firoiu and M. Borden. "A study of active queue management for congestion control", In IEEE INFOCOM2000, Tel-Aviv, Israel, March 26-30, 2000.

[12] Makul Mahajan, T. P., "The Modified Gaussian Function based RED (MGF-RED) Algorithm for Congestion Avoidance in Mobile Ad Hoc Networks". IJCA , 39-44, 2013.