Journal of Theoretical and Applied Information Technology
31st October 2018. Vol.96. No 20
© 2005 – ongoing JATIT & LLS

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

MARKOV-MODULATED BERNOULLI DYNAMIC GENTLE RANDOM EARLY DETECTION

[1]MAHMOUD BAKLIZI, [2]JAFAR ABABNEH, [3] MOSLEH M. ABUALHAJ, [4]NIBRAS ABDULLAH,[5]ROSNI ABDULLAH

[1,2] The world Islamic science & education university W.I.S.E
, Department of Computer Networks Systems, Jordan

[3] Al-Ahliyya Amman University, Amman, Jordan

[4,5] National Advanced ipv6 center, Universit Sains Malaysia, 1800, penang, Malaysia

E-mail: [1]mbaklizi@wise.edu.jo, [2]jafar.ababneh@wise.edu.jo, [3] m.abualhaj@ammanu.edu.jo,
[4]nibras@usm.my,[5]rosni@usm.my

ABSTRACT

This paper proposes an efficient modeling of the Dynamic (DGRED) algorithm using a three state Markov Modulated Bernoulli arrival process (MMBP-3) for primary period congestion detection at the router buffer. The purpose of using Markov is two folds, the first is to implement DGRED with multiple traffic classes, where in each class may have different priorities and the second is to enhance the mechanism of DGRED in stabilizing the average queue length (*aql*) between the allocated threshold values of *minthreshold* and *maxthreshold*, using variable calculating parameters, which is stored in the utilized states of Markov. The (MMBP-3) is employed to replace the conservative and widely used Bernoulli process (BP) in assessing DGRED method. Accordingly, a three dimensional discrete time Markov chain is provided to implement DGRED algorithm for three traffic classes where each dimension corresponds to a traffic class with its own parameters. The (MMBP-3) is deal with correlation and burstiness in the network traffic. The developed algorithm allows for faster response to the changes in the network, congestion-to-non-congestion and non-congestion-to-congestion, which lead to decrease packet losses and improvement of network performance. The proposed method is evaluated in comparison to the DGRED and other AQM methods, the results reveal that the proposed algorithm provides better performance compared to DGRED, RED, GRED, and Adaptive GRED in relations of delay packet loss, delay and mean queue length. The most appropriate arrival process for DGRED method is IBP for D, mql, P_L, and D_P while congestion occurs. Though, this process cannot discourse correlation. Thus MMBP-3 is the greatest choice as it detects both correlation and bursty things.

Keywords: *Congestion Control, Adaptive GRED, (RED), Performance Evaluation, MMBP-3, Simulation.*

1. INTRODUCTION

Through the quick evolution of mainframe networks and technologies of Internet, enhancing the performance of such networks to accommodate the great diversity of services and traffics is an increasing demand. Network performance is highly affected by the congestion [1-4]. Congestion happens at the buffer of the routers in case of the size of the received packets exceeds the existing network resources and the buffer can no longer grip all received packets [4-6].

Commonly, congestion participates effectively in deterioration computer network resources by cumulative the packet dropping (D*p*) and increasing the packet loss (PL) [7, 8]. In addition, congestion may cause to get more the mean queue length (*mql*) and the packets mean delay (*D*). Therefore, decreases the size of packets crossing the router, called the throughput (T) [9]. These increment and decrement in these measures are strongly related to the average queue length (*aql*). When *aql* value increases, T value increases. At the same time, D and PL increase and the router buffer overflows. By contrast, when the *aql* value is relatively small, D and T decrease. Active Queue Management (AQM) emerges with an adaptable utilization of threshold values and calculation of what so called dropping probability. Dropping probability is calculated with reference to the threshold value and the status of the queue size. AQM calculates the value of *aql* first, and then compares it with the given threshold. In case the *aql* value is greater than the threshold value, all packets that arrive at the buffer are dropped with the probability of preventing router buffer overflow [10-13]. Accordingly, in DGRED, as similar to other AQM methods, congestion is controlled with reference to the *aql*.

Unlike other AQM, DGRED [14], employs a dynamic mechanism to control the congestion in the router buffer and stabilize *aql* using a new defined value called Target *aql* (T*aql*). Moreover, DGRED also updates the parameters at the router buffer to enhance network performance. Accordingly, DGRED works dynamically to enhance the performance of the network according to the status of the traffic.

An AQM method, including DGRED, used Bernoouli process to model the incoming traffic and calculates values that control congestion based on this mode. However, bursty and correlation are most significant features to be captured in this traffic, which is not handled by Bernoouli process [15, 16]. As a router buffer is quickly effected to bursty and correlation features in the term of arrival process, MMBP is commonly used in to evaluate the robustness of the fast network because it handles the characteristics of both correlations and burstiness [17]. For that, this paper employs MMBP-3 for system performance and evaluation in DGRED.

The reaming of the paper is arranged as follows. Section 2 illustrates the previous work. Section 3 reflects an overview of the DGRED method. The proposed method is demonstrated in Section 4. Section 5 shows the simulation information and results of the proposed method. Finally, Section 6, stated the conclusions.

2. RELATED WORKS

Several studies on congestion control have been conducted [6, 18-22]. The Drop Tail (DT) method [23, 24] was proposed with the aim at controlling congestion using a fixed threshold to optimize the queuing delay. DT sets the threshold to the maximum capacity while dropping all incoming packets when the router buffers size exceeds the fixed threshold. The drawback of this method is the possibility of a rise in high packet queuing delay. DT might also be initialized by setting the threshold to a small value, in which the throughput T decreases. Accordingly, DT cannot optimize the network performance regardless of the threshold value. DT has several other drawbacks, such as increase in packet loss rate, saturation of the queue router buffer [4], and global synchronization [25].

Active Queue Management (AQM) is a set of methods proposed to overcome DT limitations discussed earlier. AQM methods usually start dropping packets in the early stages, unlike DT, which starts dropping packets after exceeding the threshold. Consequently, early congestion control allows the sources to decrease their transmitting rates early, before the router buffers are completely occupied. AQM controls the congestion in the router's buffer, improves the throughput, decreases packet queuing delay, decreases packet loss rate, and keeps the *mql* at minimum.

Enormous AQM algorithms for congestion control have been proposed, such as (RED) [25], Gentle RED (GRED) [26], Adaptive (GRED) [27], REDD1 [28], BLUE MMBP2 [29], DGRED [14], and other discrete-time queue analytical models [5-8]. RED is effectual method for congestion control [25]. RED controls the congestion earlier the router buffer completely full using the computed *aql* and two calculated thresholds values, *minthershold* and *maxthershold*. Generally, RED detects the congestion as follows: Initially, the computed *aql* is compared with the *minthershold* and *maxthershold*. If the *aql* is lesser than the *minthreshold*, no congestion arises. Thus, the router does not drop any packet. If the *aql* value is relies between two *maxthershold* and *minthershold* , all the arriving packets are dropped and the probability is calculated as the D_p to alleviate the congestion. Finally, when the *aql* is over the *maxthershold*, totally all the arriving packets are dropped at a D_p value equal to one (Figure 1).

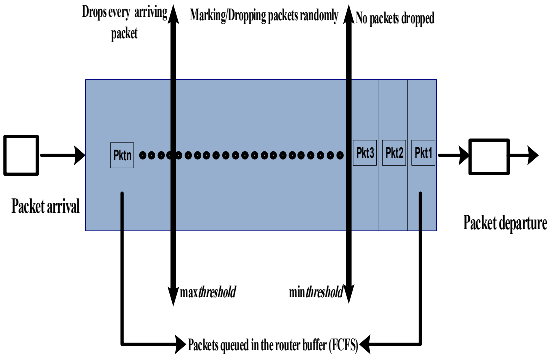


Figure 1: The RED Method Router Buffer

Generally, RED's disadvantage is that the calculated *aql* varies in accordance with the congestion case. Therefore, if the congestion case is light, the *aql* value will be near to the *minthreshold*. If the congestion case is severe, the *aql* value will be near to the *maxthreshold*; thus, D_p will increase and the buffer will overflow. Additional disadvantage is that the calculated *aql* be on the traffic load. Consequently, if the traffic load is great, *aql* value may exceed the *maxthreshold*. In such a case, network performance will be decreased in many aspects. So, the router buffer will drop all the packets arrive. Moreover, RED parameters shall be determined at certain values to achieve high performance [30, 31].

Floyd [26] the proposed GRED to overcome some limitations in RED [26, 31, 32]. Similar to RED, the main aim of the GRED algorithm is to manage the congestion networks at the primary stage. GRED uses a like method that is used by RED in calculating D_p to stabilize *aql* at a definite level. However, GRED utilizes three thresholds (*minimum*, *maximum*, and *doublemaximum*). Usually, GRED deal with the arriving packets built on one of the following situations (Figure 2):

1. When *aql* at the router is lesser the *minthreshold*, the GRED doesn't drop any packet.
2. If *aql* is between the *minthreshold* and *maxthreshold*, the GRED will drop the arriving packets similar to RED.
3. If *aql* is between the *maxthreshold* and *doublemaxthreshold*, the GRED will drop the packets with higher probability.
4. If *aql* is equal or above the *doublemaxthreshold*, The dropping value is equal to one.

Unfortunately, GRED has some disadvantages. First, GRED has more than threshold values to deal with. Second, parameterization. Third, in case that *aql* is below *minthreshold* with high high happens, *aql* takes time to adjust and the router buffers will possible overflow during the adjustment process.

AGRED is suggested to improve the GRED algorithm during congestion at the router buffer. Also, AGRED goals at improving the parameter settings (e.g., the *maxthreshold* and the maximum value of D_{int}, which is equal to D_{max} in GRED). The computation of *aql* in AGRED is also like to that in GRED. Consequently, AGRED decides on the packet dropping in a manner similar to that in GRED [27] (Figure 2).

The difference between AGRED and GRED is in the calculation of D_{init} (the initial packet dropping (D_p)). In AGRED, D_{init} value varies between D_{max} value to 0.5 as long as the *aql* value is between the *doublemaxthreshold* and *maxthreshold*. In GRED, when *aql* value is between the doublemaxthreshold and *maxthreshold*, D_{init} value varies between D_{max} value to 1.0.

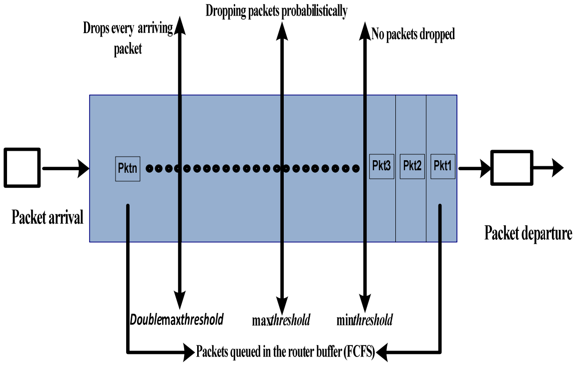


Figure 2: The GRED and AGRED Router Buffer

DGRED is another extension of GRED [14]. DGRED employs a dynamic *maxthreshold* and *doublemaxthershold* to control the congestion in the router buffer at the early stage before it overflows. The aim of the DGRED algorithm is to stabilize *aql* using a new defined value called Target *aql* (T_{aql}). T_{aql} is calculated between the *minthershold* and *maxthershold* (Figure 3). DGRED also updates the *maxthershold* and *doublemaxthershold* parameters at the router buffer to enhance network performance. DGRED uses the GRED algorithm's policy in dropping packets with probability when the *aql* is between the *minthreshold* and *doublemaxthershold*.

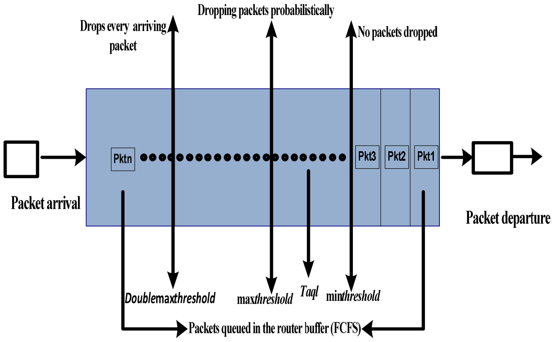


Figure 3: The Dynamic GRED Router Buffer

3. DYNAMIC GENTLE RANDOM EARLY DETECTION (DGRED)

As mentioned, the aim of the DGRED [14], is to stabilize aql using a new defined value called Target *aql* (T_{aql}). DGRED mechanism is implemented in four steps, as discussed in the following: **Step 1**: *initialization of the parameter setting* is produced during the packet incoming time. DGRED initiates the *minthreshold* and *maxthreshold* to the same values as those in the GRED and RED algorithms [25, 26]. Furthermore, the *doublemaxthershold* is set to the same significance as that in GRED [26]. The *aql* is initialized to 0.0 and the counter (C) value is set to –1. The parameter C denotes the amount of packets that reached at the buffer thus far without being dropped since the previous dropping. The value of the *aql* is initialized in this stage as well.

Step 2: *Target Value* (T_{aql}) is then calculated using Equation 1. T_{aql} value was introduced in DGRED to stabilize the *aql* between the *minthershold* and *maxthershold* and to detect congestion at the early stage. The indicated position by T_{aql} identifies the incipient congestion situation.

$$T_{aql} = \frac{minthreshold + doublemaxthershold}{\#\,threshold} \qquad (1)$$

where #threshold is refer to the number of thresholds that is used by the algorithm (*minthershold*, *maxthershold* and *doublemaxthershold*). Equation 1 is derived to get a value for T_{aql} between *mint* and *maxthershold* values. GRED recommends that the setting value of the *maxthershold* is double that of the *minthershold* [26]. Thus, any setting value for *min* and *maxthershold* can be used with Equation 1 to provide a value for T_{aql} between *minthershold* and *maxthershold* values.

Step 3: *the aql* value is computed based on whether the router buffer is empty or not, as explained in Figure 4. Consequently, in the case that the queue of the router buffer is empty, the *aql* value is computed based on the present idle time (n). The *aql* value is computed using Equation 2. So, if the buffer is not empty, the *aql* is computed using Equation 3.

$$aql = aql \times (1 - qw)^{n} \qquad (2)$$

$$aql = aql \times (1 - qw) + qw \times q\_instantaneous$$
(3)

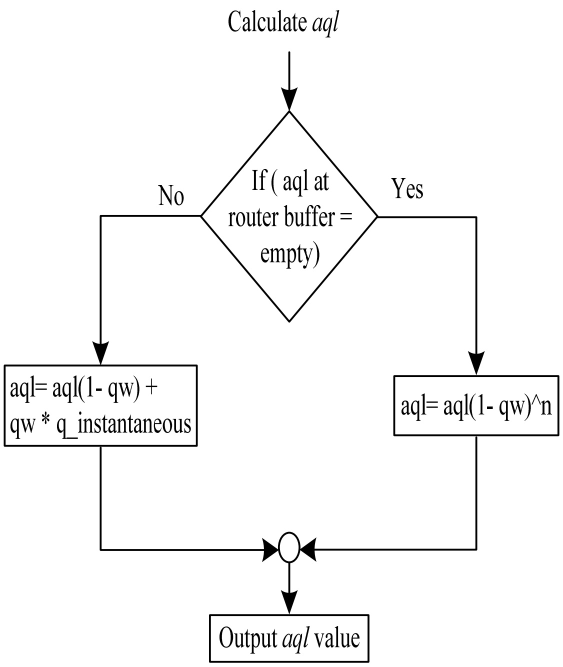


Figure 4: The process of aql Calculation

Step 4: the calculated *aql* value is compared to the value of T_{aql} and then *updates doublemaxthershold and maxthershold locations* to improve network performance (Figure 5). The values of both *doublemaxthreshold and maxthreshold* are updated with mention to the *aql*. The values of *doublemaxthreshold and maxthershold* values are decreased and increased by Equations 4 to 6 to manage and control congestion at the router buffers. This management is done by updating *doublemaxthershold and maxthershold* values by increasing and decreasing around the T_{aql} level. Thus, the *aql* value stabilizes at the T_{aql} level. This stabilization prevents router buffers from overflowing. As a result, fewer packets are dropped.

maxthd-(doublemaxthrd-minthrd) × (1/# of thrd) (4)

doublemaxthrd–(doublemaxthrd – minthrd) × (1/# of thrd) (5)

maxthrd – (doublemaxthrd – minthrd) × (1/# of thrd) (6)

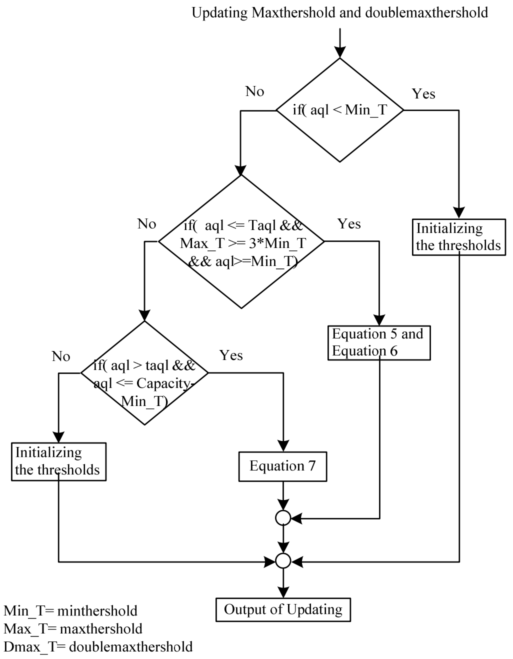


Figure 5: Updating Process of maxthershold and doublemaxthershold

So, if *aql* value is less than *minthreshold* value, no congestion happens and *maxthreshold and minthreshold* values will not be updated [26]. Though, if *aql* is avove than *minthreshold* value and equal than or less to T_{aql}, and maxthreshold value is equal than or greater to three times of *minthreshold*, then *maxthreshold* and *doublemaxthreshold* values are changed using Equations 5 and 6, respectively. As such, *aql* value rises accordingly to be stabilized at T_{aql}. On the other hand, if *aql* is greater than T_{aql} and less than or equal to the value of (capacity of buffer - *minthreshold*), *doublemaxthreshold and maxthreshold* values are updated using Equation 7. Therefore, they become the same and they avoid *doublemaxthreshold* value to go above the buffer capacity. Then, *doublemaxthershold and maxthreshold* values are increased to push *aql* toward the T_{aql} and to ease congestion in the routers buffer by processing more packets. Lastly, if nothing of these cases happens, *maxthreshold* is put to the same values as those in the RED methods [25] and *doublemaxthreshold* is put to a same value as that in GRED [26].

4. THE PROPOSED MMBP3-DGRED

BP is used for a modeling packet arrival process in discrete-time queuing systems [33, 34] with the existence of correlation and burstiness of packet arrivals is presently not a good choice [33, 34]. The performance of discrete time queuing system is extremely sensitive to correlation and burstiness in the arrival process modle.MMBP-3 [33, 34]is generally used as a source model in the performance analysis of high speed discrete time queuing networks because it can engross correlation and burstiness properties in the arrival process model[33, 34]. In this paper, 3-States Markov Modeling Bernoulli Process (MMBP3) is proposed to replace the Bernoulli process in DGRED. As bursty and correlation are most significant traffic features, Bernoulli process is not appropriate theory for an incoming packets process of the packets in a discrete times structure [15, 16]. As a router buffer is greatly sensitive to bursty and correlation features in the arrival packets process, (MMBP) is one the models that used to evaluate the robustness of the fast network traffic because it handles the characteristics of both correlations and burstiness [17]. For that, this paper employs MMBP-3 for system performance and evaluation.Time in the proposed method is discredited into fixed length slots and the process uses a geometric period of time slots for each state (Figure. 6).

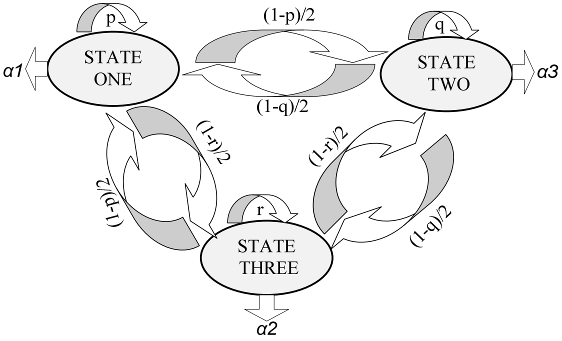


Figure 6: Three-state MMBP

In the propose method, the arrival process has three separate states and the source produces packets for traffic class one in state-one, traffic class two in state-two and traffic class three in state-three. State transitions is implemented as follows: When the arrival process is in state number one in time slot k, it produces an arrival with probability equal a1 and may still in this state in the next time slot (k + 1) with probability equal p. in case the arrival process is in the state number two in time slot k, it produces a packet with probability equal a2 and may still in state-two in the next time slot (k + 1) with probability q and finally, when the arrival process is in state number three in time slot k, it produces a packet with probability a3 and may still in state three in the next time slot (k + 1) and with probability r. The change probability from state one to state two and state three is equal to (1 – p/2), from stat two to state one and state three is equal to (1 – q/2), and the change probability from state tree to state one and state two is equal to (1 – r/2). Therefore, the probability that a time slot contains an arrival packet is a Bernoulli process with a parameter a1, a2 and a3 that differs according to a three state Markov process which is independent of the arrival distribution. So, MMBP3 is characterized by its change probability matrix P and a diagonal matrix K of arrival probabilities, Equation 7.

$$P = \begin{bmatrix} p & (1-p)/2 & (1-p)/2 \\ (1-q)/2 & q & (1-q)/2 \\ (1-r)/2 & (1-r)/2 & r \end{bmatrix}, A = \begin{bmatrix} a1 & 0 & 0 \\ 0 & a2 & 0 \\ 0 & 0 & a3 \end{bmatrix}$$

(7)

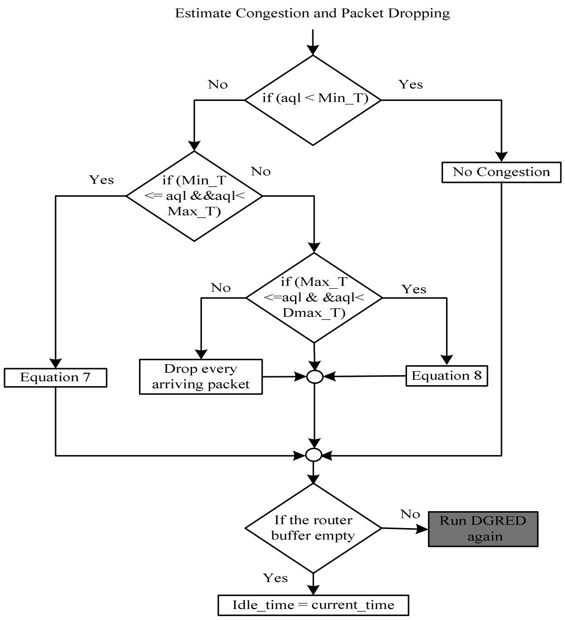
The steady state probabilities of the MMBP3 in each state can be achieved from the balance equations for this three state chain and are known by Equation 8, Equation 9 and Equation 10, respectively.

$$P(0) = pp(0) + (1-q)/2p(1) + (1-r)/2p(2) \quad (8)$$

$$P(1) = (1-p)/2p(0) + qp(1) + (1-r)/2p(2) \quad (9)$$

$$P(2) = (1-p)/2p(0) + (1-q)/2p(1) + rp(2) \quad (10)$$

where P(0), P(1) and P(2) are the steady state probabilities that the MMBP is in state one, state two, and state three separately. Therefore, the proposed MMBP3-DGRED is operated, *congestion is estimated and packet dropping is implemented*, as shown in Figure 7.



Min_T= minthershold
Max_T= maxthershold
Dmax_T= doublemaxthershold

Figure 7: Estimated and Evaluation Congestion and Packet Dropping

The status of congestion is evaluation and estimated based on *aql* value. So, if *aql* value is below than *minthreshold*, no packet is dropped since no congestion is offered at DGRED router buffer. In addition, C value is set to –1 and D_p is set to 0.0. Therefore, no packet is dropped. On the other hand, if *aql* value is between *maxthreshold* and *minthreshold* values,The DGRED router buffer drops packets in a approach similar to that exist in GRED. Dropping packets is specified with calculating D_p for the arriving packet using Equation 11and increasing C value by one.

$$D_p = \frac{\dfrac{D_{max} \times (aql - minthreshold)}{maxhreshold - minthreshold}}{(1 - C \times D_{limit})} \quad (11)$$

So, if the value of *aql* is between *doublemaxthreshold* and *maxthreshold* , DGRED algorithm router buffer drops the packets in a approach similar to that in GRED algorithm, which contains calculating D_p for the arrival packet using Equation 12 and initializing the C value to 1. Finally, if *aql* value is above than or equal to *doublemaxthreshold*, DGRED algorithm router buffer drops all arriving packet with dropp value equal one and sets C to be zero. Consequently, when the DGRED algorithm router buffer becomes empty, the idle time is set to current time.

$$D_p = \frac{D_{max} + \frac{(1 - D_{max}) \times (aql - maxthreshold)}{maxthreshold}}{(1 - C \times D_{init})}$$

(12)

5. SIMULATION AND PERFORMANCE EVALUATION

RED, GRED, AGRED, DGRED, GRED-MMBP3 and the proposed DGRED-MMBP3 are simulated and evaluated dependent on a discrete time queue that employs slot as a time [20, 35]. Every slot might include arrived or departed packets. We have used one router's buffer to implement and compare the proposed DGRED-MMBP3 method with the other methods. Whereas, the arrived or departed packet implemented in one mode. The preparation mode is first in first out . The implementation is applied in Java language on an i7 processor with 1.68 GHz and 3 GB RAM. In the showed simulation, the probability of the arriving is denoted by α [35]. *β* denotes to the probability of packet departure. The arrived packets can be demonstrated using a Bernoulli process, whereas the departed packets can be demonstrated using a geometrical distribution. Using geometrical distribution.

The performance of the proposed DGRED-MMBP3 is compared with those of DGRED, GRED, GRED-MMBP3, AGRED, and RED. The performances of these methods are measured and tested ten times in ten runs, each run using different seeds as input to the random number generator. This main step removes possible unfairness in the output results and produces assurance intervals for the performance measures in the simulation. The performances of all AQM methods are computed and evaluated when the system arrive a steady state. The buffer size consist of 20 packets was used to sense congestion at small buffer sizes in DGRED, RED, GRED and AGRED and the buffer size room of 35 packets in GRED-MMBP3 and DGRED-MMBP3. The number of slots used in the tests was 2000000. This value is sufficient warm up period, The warm up period is finished when the system arrives a steady state. The values of D_{max}, *minthreshold*, *maxthreshold*, and *qw* are fixed to 0.1, 3, 9 and 0.002, respectively in RED, GRED, DGRED and AGRED and for the first class in GREDMMBP3 and DGRED-MMBP3, as recommended in RED [25], while these values are set to 4,12, 0.1, and 0.002, respectively for the second class in GRED-MMBP3 and DGRED-MMBP3and are set to 5, 15, 0.1, and 0.002, respectively for the third class in GRED-MMBP3 and DGRED-MMBP3 [26]. Table 1 contains all the utilized parameters. The simulation evaluation results are measured using several performance metrics (e.g., *mql*, *T*, *D*, P_L, and D_p), which are discussed in the following subsection

Table 1: Parameter settings for GRED, AGRED, RED GREDMMP3 and DGREDMMP3 algorithms

Parameter	GRED, AGRED	RED	GREDMMP3,DGREDMMP3
The Probability of packets arrival	0.18-0.93	0.18-0.93	0.18-0.93
Prospect of departed packets	0.5	0.5	0.5
Buffer size	20	20	35
Qw	0.002	0.002	0.002
Dmax	0.1	0.1	0.1
Number of slots	2000000	0.1	0.1
minthreshold	3	3	3, 4 and 5
maxthreshold	9	9	9, 12 and 15
doublemaxthreshold	18	-----	18, 24, 30

5.1 Mql, Throughput, and Delay

Figure 8, Figure 9 and Figure 10 illustrate the output performances of RED, GRED, AGRED, DGRED, GRED-MMBP3 and DGRED-MMBP3 using different probabilities of packet arrivals. Specially, Figures 8 shows the mql versus the probability of packet arrival, Figure 9 illustrates delay between all the involved methods and Figure 10 illustrates throughput between all the involved methods.

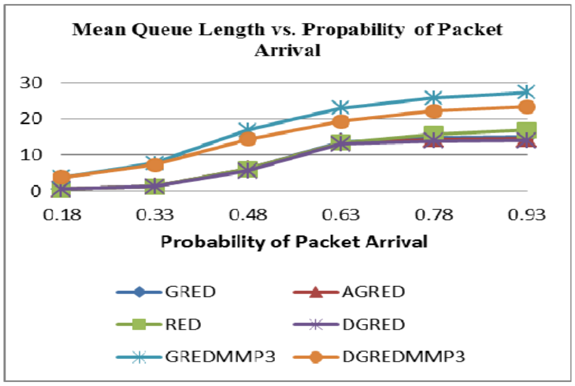


Figure 8: Results of mql vs. probability of packet arrival.

In Figure 8, *mql* for RED, GRED, AGRED and DGRED are same up to sure value of the probability of packet arrival (e.g., 0.33), with similar observation between the proposed DGRED-MMBP3 and GRED-MMBP3. In case a low probability value, there is at most a light congestion case since the probability of packet departure is greater than that of packet arrival ($a > B$). In such case, all the compared methods sustain a good and stable *mql*. However, for a higher probability value, congestion is more likely to happen at the router buffers. Accordingly, the *mql* of the AQM algorithms increases exponentially. The proposed algorithm, on the other hand, performs better than the AQM algorithms in terms of *mql* at such high probability values. This phenomenon occurs mainly because DGRED drops fewer packets than RED, GRED and AGRED and DGRED-MMBP3 drops fewer packets than GREDMMBP3.

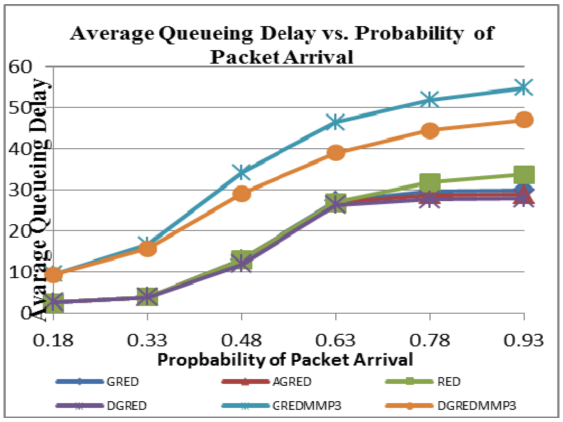


Figure 9: Results of D vs. probability of packet arrival

In Figure 9, once again, DGRED and DGRED-MMBP3 achieve better in terms of the average delay. However, AGRED also shows good performance in terms of delay. This result is due to the fewer dropped packets in DGRED than those in RED, GRED, and AGRED. Also, DGRED-MMBP3 drops fewer packets compared to GREDMMP3.

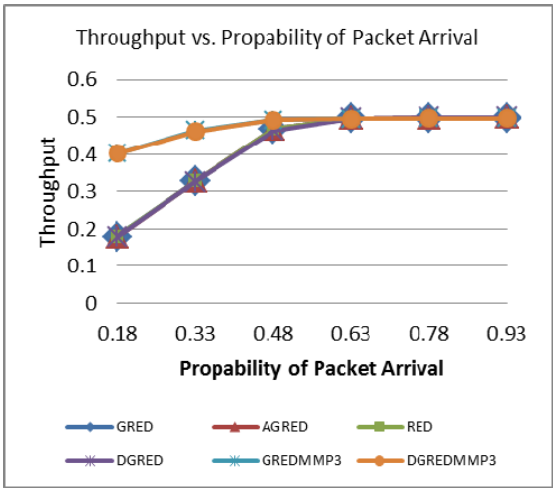


Figure 10: T vs. probability of packet arrival

Finally, in Figure 10, all the compared algorithms show similar results performance according to the throughput measure, in case that the arrived packet prospect value is less than the departed packet prospect value. Throughput results curve for the other methods are stabilized near the same value of the departed packet prospect during the congestion time.

5.2 Packet Loss and Dp

The DGRED_MMBP3 algorithm is compared with the RED, GRED, AGRED, DGRED and GREDMMP3 algorithms in terms of D_P and P_L in this section. The aim of the conducted comparison is to demonstration the quantity of packets dropping at the router buffer in all compared methods. The performances of RED, GRED, ARED, DGRED, GRED-MMBP3 and DGRED-MMBP3 methods in terms of P_L and D_P are showed in Figures 11 and Figure 12, respectively.

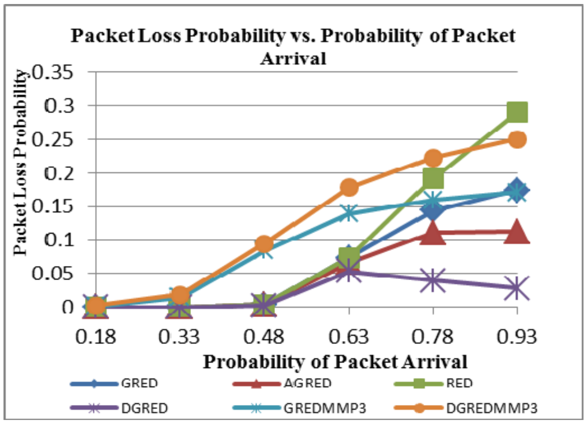


Figure 11: PL vs. probability of packet arrival

In Figures 11, the DGRED-MMBP3 and DGRED marginally obtains the best and least P_L performance when the prospect of packet departure is less than the prospect of packet arrival, since the router buffer in the DGRED-MMBP3 and DGRED algorithms overflows at an earlier time compared with those in the GRED, RED, AGRED and GRED-MMBP3 algorithms. When arrived packets prospect is less than the departed packets prospect, all algorithms obtain similar P_L results under no congestion status.

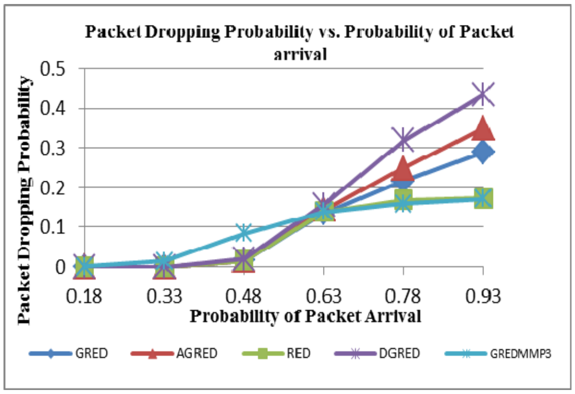


Figure12: Dp vs. probability of packet arrival

Similarly, in Figures 12, the proposed DGRED-MMBP3 and DGRED evidently drop more packets at the router buffer than the RED, GRED, AGRED and GRED-MMBP3 When arrived packets prospect is higher than the departed packets prospect. Likewise, the reason for this result is since the router buffers in the DGRED and DGREDMMP3 algorithms overflow at an earlier stage compared with those in RED, GRED, AGRED and GREDMMP3.

In summary, the DGRED-mmp3 offers satisfactory performance results when the packets arrival greater than the packets departure or less than the packet departure. Unlike AQM methods such as BLUE-MMP2. This uses two states to detect the Correlation and burstiness. In this case when the traffic is high the performance results degrade and the router buffer overflows.

6. CONCLUSIONS

Correlation and burstiness are mostly important features for heavy traffic. However, restitution traffic model as the PB processes are enable to capture Correlation and burstiness. In this paper a (MMBP3) as the traffic source for DGRED, which detects congestion at router buffers at an initial state before the router buffer arrives to the limit. DGRED-MMBP3 is compared with DGRED, GRED, RED, AGRED and GRED-MMBP3 in measures of mql, T, D, P_L, and D_p, and the observations were as follows:

- RED, GRED, AGRED, and DGRED give similar measure results at what time the of arrived packets value less than the departed packets value.
- GRED-MMBP3 and DGRED-MMBP3 give similar performance measure at what time the arrived packets prospect is reached to a value less than the departed packets prospect.
- DGRED and DGRED-MMBP3 slightly gives better mql and D results than RED, GRED, AGRED and GRED-MMBP3 at what time the values of arrived packets are higher than the values of departed packets. Also, RED, GRED, AGRED, DGRED, GRED-MMBP3 and DGRED-MMBP3 obtain similar T results with such values of packet arrival probability.
- DGRED-MMBP3 slightly outperforms the GRED-MMBP3 for P_L while the departed packets value is less than the arrived packets value.
- Moreover, GRED-MMBP3 drops fewer packets (Dp) at their router buffers compared to DGRED-MMBP3 at such values of packet arrival probability.

REFRENCES:

[1] G.Thiruchelvi and J.Raja, " A Survey On Active Queue Management Mechanisms " *IJCSNS International Journal of Computer Science and Network Security,* vol. 8, December 2008 2008.

[2] M. Welzl, *Network Congestion Control: Managing Internet Traffic*. Chichester, 2005.

[3] L. Yu, M. Chen, Y. Gong, H. Bai, and C. Xing, "Detecting Shared Congestion Paths Based on Sparse Representation," *International Information Institute,* vol. 15, pp. 323-330. , 2012.

[4] M. baklizi, J. Ababneh, and " A Survey in Active Queue Management Methods According to Performance Measures," *International Journal of Computer Trends and Technology (IJCTT),* vol. 38 p. 145, 2016.

[5] A. S. Tanenbaum, *Computer Networks*, 4 ed.: Prentice Hall Ptr, 2002.

[6] M. Baklizi and 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, 2016.

[7] Li, Jian-Min, Widjaja, Indra, Neuts, and Marcel, "detection in ATM network," *Journal of Performance Evaluation* pp. 147–168, 1998.

[8] M. BAKLIZI, J. ABABNEH, and N. ABDALLAH, "PERFORMANCE INVESTIGATIONS OF FLRED AND AGRED ACTIVE QUEUE MANAGEMENT METHODS," in

Proceedings of Academicsera 13 th International Conference, Istanbul, Turkey, 2018.

[9] D. Lin and R. Morris, "Dynamics of random early detection," in *ACM SIGCOMM*, New York, 1997, pp. 127-137.

[10] A. Bitorika, M. Robin, M. Huggard, and C. M. Goldrick, "A Comparative Study of Active Queue Management Schemes," in *Proceddings of IEEE ICC 2004, Congestion Control Under Dynamic Weather Condition 103*, 2004.

[11] Ryu.S, "Active Queue Management (AQM) based Internet Congestion Control," University at Buffalo, October 1, 2002.

[12] J. H. Salim, U. Ahmed, and J. 2000, "Performance evaluation of explicit congestion notification (ECN) in IP networks," N. W. Group, Ed.: RFC 2884, 2000.

[13] m. baklizi, H. Abdel-Jaber, A. A. Abu-Shareha, M. M. Abualhaj, and S. Ramadass, "Fuzzy Logic Controller of Gentle Random Early Detection Based on Average Queue Length and Delay Rate " *International Journal of Fuzzy Systems,* vol. 16, 2014.

[14] M. Baklizi, H. Abdel-jaber, M. M. Abu-Alhaj, N. Abdullah, S. Ramadass, and A. ALmomani1, "DYNAMIC STOCHASTIC EARLY DISCOVERY: A NEW CONGESTION CONTROL TECHNIQUE TO IMPROVE NETWORKS PERFORMANCE," *International Journal of Innovative Computing, Information and Control,* vol. 9, 2013.

[15] D. L, A. .I.U, Woodward.M.E., and W. Xingang, "Discrete-time performance analysis of a congestion control mechanism based on RED under multi-class bursty and correlated traffic," *The Journal of Systems and Software,* pp. 1716–1725, 2007.

[16] Y. Li, J. Xu, and K. Lu, "Bi-modal Multi-class Uniformly Pareto-improving Congestion Pricing and Revenue Refunding," *international Information Institute,* vol. 15, pp. 2543-2554, 2012.

[17] Nilsson and Cui, "The impact of correlation on delay performance of high speed networks.," *Southeastern Symposium on system Theory,* 1994.

[18] J. Ababneh, H. Abdel-Jaber, F. Thabtah, W. Hadi, and E. Badarneh, "Derivation of Three Queue Nodes Discrete-Time Analytical Model Based on DRED Algorithm," in *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, 2010, pp. 885-890.

[19] H. Abdel-jaber, M. Woodward, F. Thabtah, and M. Al-diabat, "Traffic Management for the Gentle Random Early Detection Using Discrete-Time Queueing," in *Proceedings of the International Business Information Management Conference* Marrakech, Morocco, 2008, pp. 289-298.

[20] H. Abdel-Jaber, M. Woodward, F. Thabtah, and A. Abu-Ali, "Performance evaluation for DRED discrete-time queueing network analytical model," *Journal of Network and Computer Applications,* vol. 31, pp. 750-770, 2008.

[21] H. Abdel-jaber, M. Woodward, F. Thabtah, and M. Al-diabat, "Modelling BLUE Active Queue Management Using Discrete-Time Queue," in *Proceedings of the 2007 International Conference of Information Security And Internet Engineering (ICISIE'07)*, London, 2007, pp. 568-573.

[22] M. T. a. S. S. Ashraf Uddin Ahmed, "A New Approach on Zone-based Routing Algorithm for Congestion Control Over the Internet," *International Information Institute,* vol. 9, 2006.

[23] C. Brandauer, G. Iannaccone, C. Diot, and S. Fdida, "Comparison of Tail Drop and Active Queue Management Performance for bulk-data and Web-like Internet," in *IEEE ISCC 2001*, 2001, pp. 122-129.

[24] R. Stanojevic, R. N. Shorten, and C. M. Kellett, "Adaptive tuning of drop-tail buffers for reducing queueing delays," *Communications Letters, IEEE,* vol. 10, pp. 570-572, 2006.

[25] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," in *Networking, IEEE/ACM Transactions on*, 1993, pp. 397-413.

[26] S. Floyd, "Recommendations On Using the Gentle Variant of RED," 2000.

[27] M. Baklizi, H. Abdel-jaber, S. Ramadass, N. Abdullah, and M. Anbar, "Performance Assessment of AGRED, RED and GRED congestion control algorithms,"

Information Technology Journal, vol. 11, pp. 255-261, 2012.

[28] H. Abdel-jaber, F. Thabtah, A. M. Daoud, J. Ababneh, and M. Baklizi, "Performance Investigations of Some Active Queue Management Techniques Using Simulation " *International Journal on New Computer Architectures and Their Applications,* vol. 2, 2012.

[29] Adeeb alsaaidah, mohd Zalisham, Mohd Fadzli , and H. Abdel-jaber, "Markov-Modulated Bernoulli-Based PErformence Analysis for Gentle Blue and Blue Algorithms under Bursty and Correlated traffic," *Journal of Computer Sciences,* vol. 12, pp. 289-299, 2016.

[30] F. Wu-chang, K. G. Shin, D. D. Kandlur, and D. Saha, "The BLUE active queue management algorithms," *Networking, IEEE/ACM Transactions on,* vol. 10, pp. 513-528, 2002.

[31] 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 August 1, 2001 2001.

[32] J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Computer Networks,* vol. 36, pp. 203-235, 2001.

[33] N. Chee-Hock and B.-H. Soong, *Queueing Modelling Fundamentals: With Applications in Communication Networks*: Wiley Publishing, 2008.

[34] C. H. Ng, L. Yuan, W. Fu, and L. Zhang, "Research note: Methodology for traffic modeling using two-state Markov-modulated Bernoulli process," *Comput. Commun.,* vol. 22, pp. 1266-1273, 1999.

[35] M. E. Woodward, *Communication and computer networks: Modelling with discrete-time queues*. London: IEEE Computer Society Press (Los Alamitos, Calif.), 1994.