International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
Available at http://inpressco.com/category/ijcet

Research Article

Performance Evaluation of the Proposed Enhanced Adaptive Gentle Random Early Detection Algorithm in Congestion Situations

Mahmoud Baklizi[#*] **and Jafar Ababneh**[#]

[#]Information Systems and networks, The World Islamic Sciences & Education University (WISE), Amman-Jordan

Accepted 05 Sept 2016, Available online 07 Sept 2016, **Vol.6, No.5 (Oct 2016)**

Abstract

Congestion control methods are continuously linked with the rapid advances in Internet and network technology. Congestion generally occurs when the amount of packets arriving at the router buffer cannot be accommodated. This paper proposed an Enhanced Adaptive Gentle Random Early Detection (Enhanced AGRED) method based on Adaptive Gentle Random Early Detection (AGRED) method in order to detect the congestion in early stage before the router buffer overflows by enhancing the parameter setting of Queue Weight (Qw). The Enhanced AGRED is simulated and compared with the AGRED and Gentle Random Early Detection (GRED) methods. The simulation results for the proposed Enhanced AGRED, GRED and AGRED methods are carried out by varying the variable of packet arrival probability to create different congestion/non-congestion scenarios. During the congestion, the simulation results reveals that Enhanced AGRED offers marginally better performance results than GRED and AGRED, with regard to mean queue length, average queuing delay and packet loss probability due to overflow. Therefore, the results prove that Enhanced AGRED is an effective method in controlling congestion router buffers of networks. Whereby, improve networks performance.

Keywords: *Congestion Control, Active Queue Management, discrete-time-queue, performance evaluation.*

1. Introduction

The Internet technologies and the performance of computer networks are constantly being questioned because of their rapid development. Congestion is one of the main problems that challenge network performance (G.Thiruchelvi and J.Raja, 2008, Welzl, 2005). Congestion occurs when the incoming packets to the router buffers can no longer handle the incoming packets, as the amount of incoming packets exceeds the available network resources (Tanenbaum, 2002). The drawbacks of congestions are as follows. Congestion plays a major role in worsening network performance by increasing the packet dropping probability (Dp) and growing the packet loss probability (PL). In addition, congestion may lead to an increase in the mean queue length (mql) and the mean waiting time (D) of packets, which will finally degrade the amount of packets passing through the buffer of the routers, namely, the throughput (T) (Lin and Morris, 1997).

Active queue management methods have been proposed to improve the network performance (Baklizi *et al*., 2013, Baklizi *et al*., 2012, Abdel-jaber *et al*., 2008). Many researchers have been proposed an AQM method, such as (Kiruthiga and Raj, 2014, Das *et al*., 2013, Singh and Balveer, 2013), which were proposed to overcome the limitations of the DT method discussed earlier (Bitorika *et al*., 2004, Salim and Ahmed, 2000). Enormous methods for congestion control have been built as AQM, such as Random Early Detection (RED) (Floyd and Jacobson, 1993), Adaptive Random Early Detection (ARED) (Floyd *et al*., 2001), Random Exponential Marking (REM) (Athuraliya *et al*., 2001, Lapsley and Low, 1999) , BLUE (Feng *et al*., 1999, Feng *et al*., 2002), Stochastic Fair BLUE (SFB) (Feng *et al*., 2001), Gentle Random Early Detection (GRED) (Floyd, 2000), Dynamic Random Early Drop (DRED) (Aweya *et al*., 2001), Stabilized Random Early Drop (SRED) (Ott *et al*., 1999), Fuzzy BLUE (Yaghmaee and AminToosi, 2003), Fuzzy Exponential Marking (FEM) (Chrysostomou *et al*., 2003), Decbit (Ramakrishnan and Raj, 1988), Enhanced Random Early Detection(ENRED) (Alshimaa *et al*., 2014), Adaptive Neuro Fuzzy Inference System (ANFIS) (Kusumawardani, 2013), AGRED (Baklizi *et al*., 2012), and DGRED (Baklizi *et al*., 2013) .

One of the most known AQM method is The Random Early Detection (RED), was proposed by Floyd and Jacobson in 1993.

RED uses *aql* and two calculated thresholds values, namely, minthreshold and maxthresholdas congestion indicators (see Figure 1).

---

*Corresponding author: **Mahmoud Baklizi**

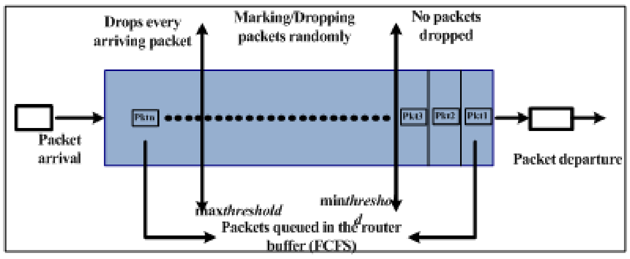

Fig 1 The single router buffer for RED

The congestion is controlled using various scenarios as follows: First, when aql is smaller than the *minthreshold*, no packets are dropped as congestion does not occur in this case. Second, if the aql is between the two thresholds, the arriving packet is dropped with calculated probability D*p* to alleviate congestion before the buffer overflowed. Finally, when the *aql* is above the *maxthreshold*, all arriving packets are dropped. By other means, the dropping probability Dp value is set to one (Figure 1).

RED is one of the most significant methods as it has been successfully adopted by the Internet Engineering Task Force (IETF) in RFC 2309 (Braden *et al.*, 1998). The advantage of RED is the elimination of the global synchronization problems (Fan *et al.*, 2012). However, RED's drawback is that the congestion indicator is embodied in computing *aql* based on the traffic load (number of connections), not on the actual status of the packet load, which may degrade the network performance in many aspects. Such as, delay and packet loss. Subsequently, RED cannot stabilize its *aql* value between the *minthreshold* and *maxthreshold* when the traffic load changes suddenly (i.e., bursty traffic) (Feng *et al.*, 2002, Floyd *et al.*, 2001, ABDEL-JABER *et al.*, 2014).

The GRED method was proposed to deal with the aforesaid limitations of RED method. Also, GRED improves the process of setting *maxthreshold* and D_{max} parameters and it is able to stabilize *aql* at a position named *Taql*. *Taql* position is half way from the main threshold and *maxthreshold* positions. Which prevents the router buffer from building its size and becoming larger than *maxthreshold* position and as a result fewer packets are dropped.

In this paper, a new method called Enhanced AGRED based on AGRED aimed to enhance the performance of AGRED with reference to the performance measures (*mql,* D, P_l) especially during heavy congestion. The queue weight (Q*w*) value of the Enhanced AGRED method varies from Qw to 2* Q*w* in all the process of computing the average queue length in the router buffer. This enables the proposed method to provide further enhancements in setting the input parameters

Gentle Random Early Detection (GRED), an adaptive method, was proposed by Floyd to overcome the limitations of RED(Floyd, 2000). GRED uses the same congestion indicator technique as RED. GRED stabilizes the *aql* at a certain level using three thresholds, namely, *minthreshold*, *maxthreshold*, and *doublemaxthreshold*.

GRED uses a new threshold value called *doublemaxthreshold* and introduces different probabilistic dropping rates between these thresholds. This makes GRED better than RED in stabilizing *aql* because when *aql* exceeds *maxthreshold* a higher probability is used to prevent buffer overflow. While the stabilization mechanisms of GRED and RED are different, calculating Dp in GRED is partially similar to the one in RED. Generally, GRED reacts with the arriving packets based on one of the following scenarios (Figure 2).

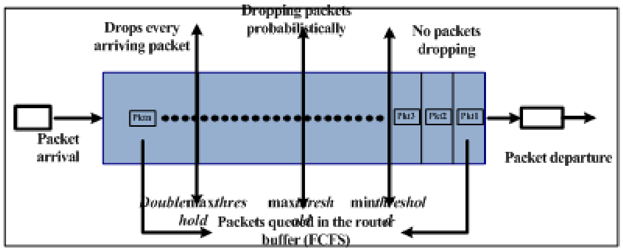

Fig 2 The single router buffer for GRED

- When *aql* at the router is below the *minthreshold*, no packets are dropped.
- When aql is between the *minthreshold* and *maxthreshold*, the router will drop the arriving packets randomly, similar to RED.
- When *aql* is between the *maxthreshold* and the *doublemaxthreshold*, the packets are dropped randomly with a higher probability compared to the previous case.
- When *aql* is equal or greater than the *doublemaxthreshold*, the GRED router drops the arriving packets with Dp equal to one (i.e., arriving packets are dropped).

Unfortunately, GRED has some limitations. First, GRED deals with several threshold values and must set its parameters to specific values to obtain satisfactory performance (i.e., parameterization). Second, when the *aql* is less than the *minthreshold* and heavy congestion occurs, the *aql* will take time to adjust, during which the router buffer will likely overflow. Thus, no packets are dropped despite the overflowing GRED of router buffer(Aweya *et al.*, 2001, Floyd *et al.*, 2001, Floyd, 2000).

Adaptive Gentle Random Early Detection (GRED), an adaptive method, was proposed by Floyd to overcome the limitations of GRED (Baklizi, 2012). AGRED uses the same congestion indicator technique as GRED. The way of calculating the Dint in AGRED method according to the following Equation

$$D_{int=}\ D_{max}\ +\frac{(1-D_{max})}{2}\times\frac{(aql-maxthreshold)}{maxthreshold} \qquad (1)$$

In AGRED method, when the average queue length is between *maxthreshold* and double max threshold positions, the Dint value increases from D_{max} value to 0.5 as long as the *aql* increases from max threshold to double max threshold position. This further improves the parameter settings of max threshold and D_{max} than those of GRED method.

AGRED reacts with the arriving packets based on one of the following scenarios (Figure 3).

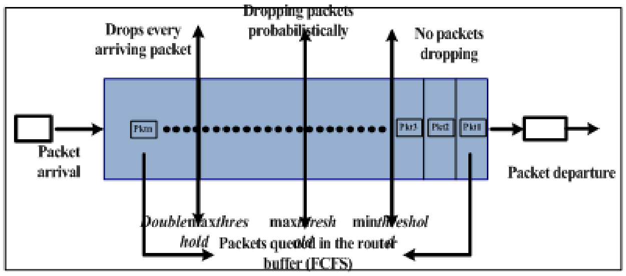

Fig 3 The single router buffer for AGRED

-When *aql* at the router is below the *minthreshold*, no packets are dropped.

- When *aql* is between the *minthreshold* and *maxthreshold*, the router will drop the arriving packets randomly, similar to RED.
- When *aql* is between the *maxthreshold* and the *doublemaxthreshold*, the packets are dropped randomly with a higher probability compared to the previous case.

-When aql is equal or greater than the *doublemaxthreshold*, the GRED router drops the arriving packets with Dp equal to one (i.e., arriving packets are dropped).

Unfortunately, AGRED has some limitations. First, AGRED deals with several threshold values and must set its parameters to specific values to obtain satisfactory performance (i.e., parameterization). Second, when the *aql* is less than the *minthreshold* and heavy congestion occurs, the *aql* will take time to adjust, during which the router buffer will likely overflow. Thus, no packets are dropped despite the overflowing GRED of router buffer (Aweya *et al*., 2001, Floyd *et al*., 2001, Floyd, 2000).

2. The proposed Enhanced AGRED

The proposed Enhanced AGRED is an extension of AGRED. Enhanced AGRED employs a dynamic value of Q*w* to control the congestion in the router buffer at the early stage before it overflows. The aim of the Enhanced AGRED algorithm is to increase the router buffer response to the dropping.

Another aim for the proposed Enhanced AGRED is providing better performance results than other AQM algorithms such as GRED and AGRED. These better performance results are represented by the results of mean queue length, average queueing delay and packet loss probability when heavy congestion has occurred.

The congestion measure of the Enhanced AGRED is aql and its parameters are similar to those of AGRED and GRED see table 1.

Table 1 Adscription of parameter used

Definitions	Description
current time	The current time.
idle time	The beginning waiting time at the Router buffer.
N	The number of packets transmitted to the router buffer through an idle Interval time.
C	A counter that represents the number of packets arrived at the router buffer and have not dropped since the last Packet was dropped.
D*p*	The packet dropping probability.
D_{init}	The initial packet dropping probability.
q_ instantaneous	The instantaneous queue length.
Q*w*	The queue weight.
D_{max}	The maximum value of $D_{init.}$
q(time)	The linear function for the time.
T*aql*	target level for the *aql*
doublemaxthreshold	is set to 2 x *maxthreshold*

Alg 1 Proposed Technique

```
Begin
 SET C = -1, aql = 0.0 // Initialization stage
FOR every arriving packet at a GRED router buffer, do //2nd Stage
    Calculate the aql for the arriving packet at the router buffer.
    Examine the queue status at the router buffer (e.g. empty or not
        IF, the queue at the router buffer = = empty, do
  Compute n, where n = q(current _ time - idle _ time)
            Set aql = aqlx (1 –qw*2) n
  ELSE
            Set aql = aqlx (1 –qw*2) + qw*2x q_instantaneous
        End IF
    END FOR
    Check the congestion status at the router buffer //3rd stage
IF, aql<min_threshold, do
SETD  = 0 .0; // No packets have dropped
        SET C = — 1;
    ELSE IF min threshold <aql&&aql<max threshold, do
SETC = C +1;
        CalculateD $_{init=}\frac{D_{max}\times(aql-\min threshold)}{\max threshold-\min threshold}$
```

```
        Calculate $D_P = \frac{D_{init}}{(1-C\times D_{init})}$
        Drop arriving packet probabilistically in terms of its$D_P$value;
        SETC = 0 ;
    ELSE IF max threshold ≤ aql&&aql< double max threshold, do
        SET C = C +1;
        SET $D_{init} = D_{max} + \frac{\frac{(1-D_{max})}{2}\times(aql-\text{max threshold})}{\text{max threshold}}$
        SET $D_P = \frac{D_{init}}{(1-C\times D_{init})}$
        Drop arriving packet probabilistically in terms of its ($D_P$)value;
        SETC = 0 ;
    ELSE IF aql≥ double max threshold, do
        Drop every arriving packet with$D_P$= 1;
        SETC = 0 ;
    END IF
    IF GRED router buffer becomes empty, do// 4th stage
        SETidle time = current time ;
    END IF
END
```

The Enhanced AGRED decides whether to drop every arriving packet as in the AGRED method figure 3.

The main difference between AGRED and the Enhanced AGRED is in calculating the *aql* value. the Enhanced AGRED computes the *aql* according to the following equations:

$$aql = aqlx\,(1 - qw * 2)\,n \tag{2}$$

$$aql = aqlx\,(1 - qw*2) + qw*2x\,q\_instantaneous \tag{3}$$

in AGRED method the value of qw is set to 0.002, which represents the queue weight similar to GRED method. This is because if qw, the queue weight, cannot be too high because the average queue size can grow quickly. While, if qw is too low, then the average queue size will respond too slowly. Thus, the value of 0.002 was chosen and also was validated in AGRED method (Madipelli *et al*., 2009). Unfortunately, the qw fixed value not appropriate in the case, that the AGRED method contains three threshold and two decision dropping scenario, in AGRED and GRED method, when the heavy congestion occur the router buffer built its size quickly and the GRED and AGRED decide to drop the packets later when the *aql* reach to the *doublemaxthreshold*. So the buffer becomes full and every packet will be drop. On the other hand, according to the above two equations, the value of qw not like GRED and AGRED fixed to the specific value equal 0.002. the equation number one used when the router buffer is empty and there is no dropping in this case. But, in equation two the value of qw plays a main role to quick response the *aql* to grow quickly to reach the *maxthreshold* and *doublemaxthreshold* to make dropping. On the other hand, the value of queue length is fix not equal the value of *aql*. In this scenario the Enhanced AGRED method keep the buffer small as soon as possible to prevent the router buffer overflow quickly.

3. Simulation information of GRED, AGRED and Enhanced AGRED

The queuing discipline, in all the simulations, is First Come First Serve (FCFS). The number of node is one, as the AQM words on the level of a single router. The bandwidth is represented as the probabilities of packet arrival and departures at the node utilized. The packets are sent to the router buffer and depart from the router buffer individually, i.e. packet by packet. The process that models the arriving packets is a Bernoulli, $a_n \in \{0, 1\}$, n= 0,1,2,3, …., where an denotes the number of packet arrivals to the buffer in slot n (Boucherie and Dijk, 2011, Neely *et al*., 2008). Bernoulli process is suitable when the buffer has a fixed length slot (Perros and Elsayed, 1994). Figure 4 illustrates the packet arrival and departure in the router buffer. The proposed methods have been simulated based on a discrete-time approach. In addition, the packets depart from the router buffer with geometrically distributed service times (Fiems *et al*., 2004). The simulation architecture is illustrated in Figure 4

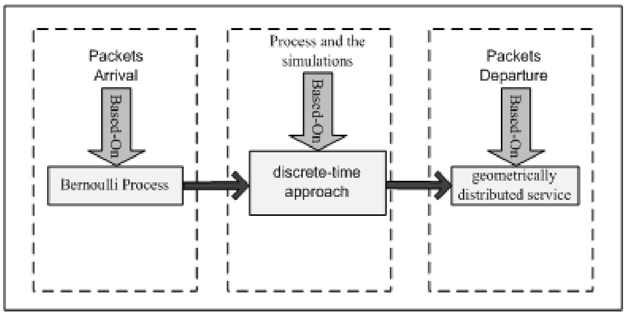

Fig.4 The packet arrival and departure in the simulation

After the simulation architecture is built, GRED, AGRED and Enhanced AGRED methods are implemented to obtain the results.

3. Performance evaluation results

The performance of the proposed Enhanced AGRED method is measured empirically, and compared with GRED and AGRED, which were discussed in related work. Ten different runs are carried through; each run is based on different seed as input to the random number generator. Thus, each run has different input numbers. Doing so, eliminates the possible bias in the output results and produces confidence intervals for the performance measures.

The performances of all the compared methods are calculated after the system reaches a steady state (Thiruchelvi and Raja, 2008). Thereby, the results have been collected as the seed slot arrives to 1100000.For the parameter settings, GRED, AGRED, and Enhanced AGRED are initiated using identical parameters. To create congestion and non-congestion scenarios at the buffer, the probability of packet arrival was set to several values. The buffer size room of 20 packets was used to detect congestion at small buffer sizes. The total time slot utilized was 2000000. This value allows for incorporating an accurate performance measure and sufficient warm-up period. The warm-up period is terminated when the system reaches a steady state. The *minthreshold*, *maxthreshold*, and D_{max}, values are set to 3, 9, and 0.1, respectively, as recommended in RED(Floyd and Jacobson, 1993). Finally, the *doublemaxthreshold* value is set to 18 as recommended in GRED (Floyd, 2000). Table 2 lists all the utilized parameters.

The simulation results are measured using several performance metrics, these are: *mql*, *T*, *D*, P_L, and Dp.

Table 2 Parameter settings for GRED, AGRED, and Enhanced AGRED methods

Parameter	GRED, AGRED, and Enhanced AGRED
Probability of packet arrival	0.18-0.93
Probability of packet departure	0.5
Router buffer capacity	20
Q_w	0.002
D_{max}	0.1
Number of slots	2000000
Minthreshold	3
Maxthreshold	9
Doublemaxthreshold	18

3.1 Mean Queue Length (Mql)

The output performance of GRED, AGRED, and Enhanced AGRED are measured with several packet arrival probabilities. Figure 5 illustrates the *mql*-based performance of the compared methods and the proposed Enhanced AGRED method versus the probability of packet arrival.

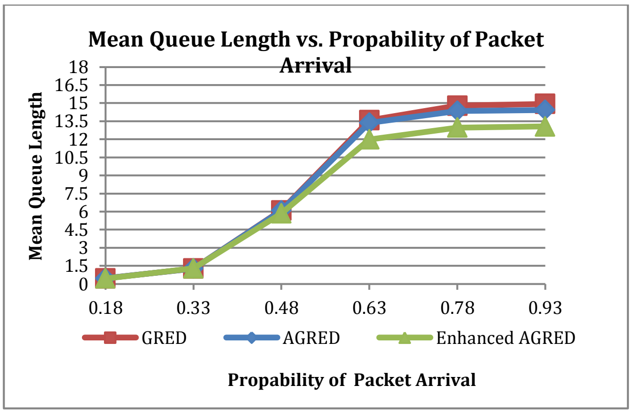

Fig.5 *Mql*-based performance vs. probability of packet arrival

The *mql* is important metric in the router buffer. Low *mql* prevents the router buffer queue from building up its size and avoids congestion as possible. As illustrated in Figure 5, *mql* for all the compared methods and the proposed Enhanced AGRED method is identical up to a certain probability value (e.g., 0.33). In such a low probability value, there is a light congestion state, as the probability of packet arrival is lower than that of packet departure (a <B), this means each packet arrives will be departed. For a higher probability values, at which congestion is more likely to exist at the router buffers, the *mql* of all the compared methods and the proposed Enhanced AGRED method increases. However, the proposed Enhanced AGRED performs better than the compared methods in terms of *mql* at such high probability values. Generally, this advantage of the proposed Enhanced AGRED is due to the fact that Enhanced AGRED overflow at times less than those of other method, which keeps the mean queue length low to serve more packets in the router buffer, and not build it size quickly.

3.2 Average Queuing Delay (D)

The average queuing delay (D), as mentioned before, is defined as the average waiting time for packets at the router buffer queue before departure. Figure 6 illustrates the delays of the compared methods and the proposed Enhanced AGRED method.

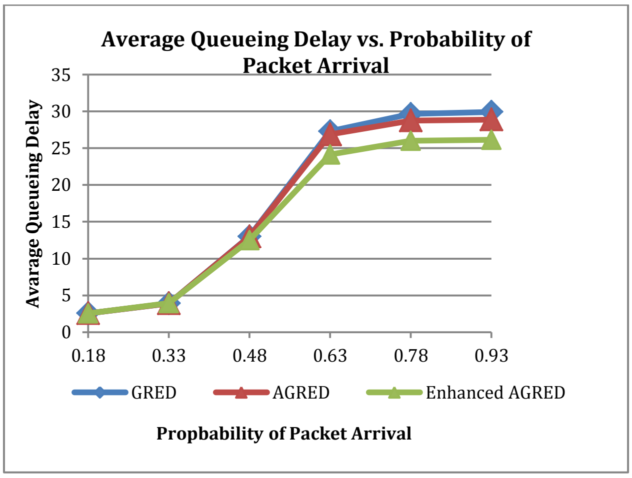

Fig.6 D-based performance vs. probability of packet arrival

Once again, the proposed DGRED performs better in terms of the average queuing delay. This is because; packets depart, in Enhanced AGRED, at times less than those for other methods. In addition, *mql* metric effects D and plays a main role in computing D. As mentioned before *D*= (*mql* /*T*). Thus, when mql is low, D is also low (Woodward, 1993).

3.3 Throughput (T)

Figure 7 illustrates the throughput-based performance under different packet arrival probabilities. The throughput of the proposed Enhanced AGRED method and the compared methods give similar T results, whether the probability of packet arrival is set to a value lower or higher than the probability of packet departure value. Figure 7 indicates that a packet arrival probability lower than the packet departure probability results in increasing T for the compared methods and the proposed Enhanced AGRED method, as long as the packet arrival probability increases up to a certain value (e.g.: 0.5). Then, all compared methods and the proposed Enhanced AGRED method, stabilize T when the packet arrival probability go above 0.5. At this value, the congestion happened in the router buffer. See Equation 5.1 (Woodward, 1993).

$$T = (1 - P[0]) * 0.5 \qquad (4)$$

where P[0], represents the probability of the packet in the router buffer.

Subsequently, when the congestion occurs, the router buffer will have no more space for any new arrival packet. In this case, the probability of the packet arrival in the router buffer is zero. Thereby, based on equation 7, T will be equal to 0.5 for all compared methods and the proposed Enhanced AGRED method.

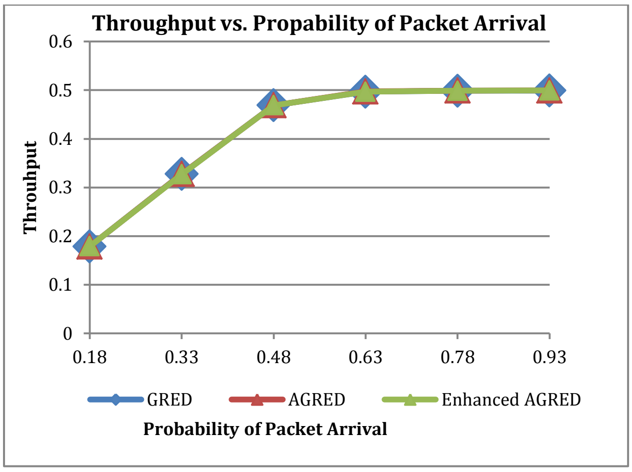

Fig.7 T-based performance vs. probability of packet arrival

3.4 Packet Loss and Dp.

The goal of the packet loss and D_p metric-based comparison is to show the quantity of packets dropping at the router buffer in all the compared methods and the proposed Enhanced AGRED method. The performances of GRED, AGRED, and Enhanced AGRED methods in terms of PL and DP are illustrated in Figures 8 and 9, respectively.

In Figure 8, the proposed Enhanced AGRED method produces the best and least P_L, when the probability of packet arrival is greater than the probability of packet departure (existence of congestion). This is because the router buffer in the Enhanced AGRED method overflows at time less than those in the GRED, and AGRED.This is because the proposed Enhanced AGRED prevents the router buffer from building its size and overflow early. When the packet arrival probability is less than the packet departure probability, all the compared methods provide similar P_L results under either a light congestion or no congestion situation

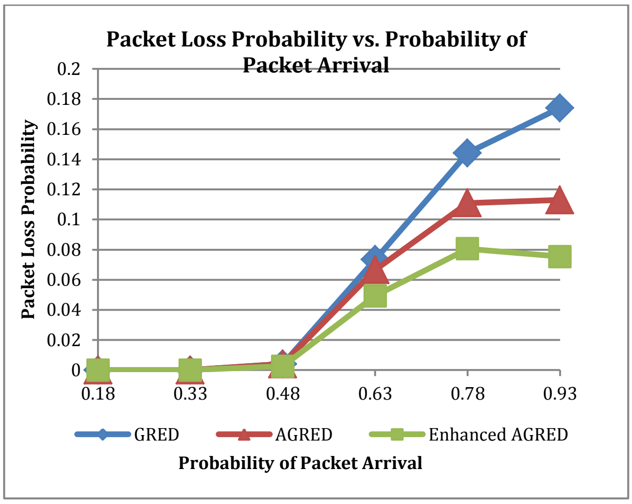

Fig.8 PL-based performance vs. probability of packet arrival

Similarly, in Figure 9, the Enhanced AGRED method evidently drops more packets at the router buffer than the GRED, and AGRED methods when the probability of packet arrival is higher than the probability of packet departure. Similarly, the reason for this result is because the router buffer in the Enhanced AGRED method overflows at an earlier time compared with those in GRED and AGRED.

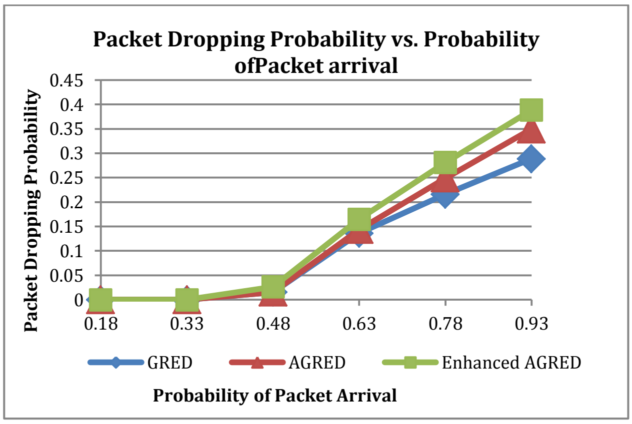

Fig.9 Dp-based performance vs. probability of packet arrival

Conclusion and future work

A suitable parameter tuning for Adaptive Gentle Random Early Detection (AGRED) was implemented in this paper. The analysis of the previous utilization of the parameter setting in AGRED shows that the inaccurate carrying of parameters' values from similar and earlier work does not give the best results. Subsequently, this approach, AGRED, required careful setting. In this paper, this setting has been accomplished and the results shows clear enhancement over the results. In the future work, we plan to investigate the utilization of different thresholding mechanisms besides considering tuning the parameters over and over to suits any modification on this method.

References

Abdel-Jaber, H., Thabtah, F. & Woodward, M. (2008) Traffic Management for the Gentle Random Early Detection using Discrete-Time Queueing. *Information Management in Modern Organizations: Trends & Challenges*

Abdel-Jaber, H., Thabtah, F., Woodward, m., jaffar, a. & bazar, H. A. (2014) Random Early Dynamic Detection Approach for Congestion Control *Baltic J. Modern Computing,* 2**,** 16-31

Alshimaa, I., Ayman, E.-S.,Zeiad, E. & Z, M. I. (2014) Enhanced Random Early Detection (ENRED). *International Journal of Computer Applications,* 92.

Athuraliya, S, Low, S. H., Li, V. H. & Qinghe, Y. (2001) REM: active queue management. *Netwrk. Mag. of Global Internetwkg.,* 15**,** 48-53.

Aweya, J., Ouellette, M. & Montuno, D. Y. (2001) A control theoretic approach to active queue management. *Comput. Netw.,* 36**,** 203-235.

Baklizi, M. (2012) Performence Assesment of AGRED, RED, and GRED Congestion Control Algorithms. *information technology journal*.

Baklizi, M., Abdel-Jaber, H., Abu-Alhaj, M. M., Abdullah, N., Ramadass, S. & Almomani, A. (2013) Dynamic Stochastic Early Discovery: A New Congestion Control Technique to Improve Networks Performance. *International Journal of Innovative Computing, Information and Control* 9**,** 1-10.

Baklizi, M., Jaber, H. A.-., Ramadass, S., ABdullah, N. & ANBAR, M. (2012) Performance Assessment of AGRED, RED and GRED Congestion Control Algorithms. *Information Technology Journal,* 11**,** 255-261.

Bitorika, A., Robin, M., Huggard, M. & Goldrick, C. M. (2004) A Comparative Study of Active Queue Management Schemes. *Proceddings of IEEE ICC 2004, Congestion Control Under Dynamic Weather Condition 103.*

Boucherie, R. J. & Dijk, N. M. (2011) *Queueing Networks: A Fundamental Approach* New York.

Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J. & Zhang, L. (1998) *Recommendations on Queue Management and Congestion Avoidance in the Internet*, RFC Editor.

Chrysostomou, C., Pitsilli Des, A., Hadjipollas, G., Sekercioglu, A. & POLYCARPOU, M. (2003) Fuzzy Explicit Marking for Congestion Control in Differentiated Services Networks. *Proceedings of the Eighth IEEE International Symposium on Computers and Communications.* IEEE Computer Society.

Das, V., Chaba, Y., Prakash, V. S., George Amalarethinam, D. I. & RAJ, E. G. (2013) Extended Queue Management Backward Congestion Control Algorithm. *Mobile Communication and Power Engineering.* Springer Berlin Heidelberg.

Fan, X., Zhang, J., Guan, L. & Wang, X. (2012) Qblue: A New Congestion Control Algorithm Based on Queuing Theory. *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on.*

Feng, W.-C., D., K. D., D., S. & G., S. K. (2001) Stochastic fair blue: a queue management algorithm for enforcing fairness. *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.*

Feng, W.-C., G., S. K., D., K. D. & D., S. (2002) The BLUE active queue management algorithms. *Networking, IEEE/ACM Transactions on,* 10**,** 513-528.

Feng, W.-C., Kandlur, D. D., Saha, D. & Shin, K. G. (1999) BLUE: A New Class of Active Queue Management Algorithms. UM CSE-TR-387-99 ed., University of Michigan, Ann Arbor, MI, Technical Report.

Fiems, D., Steyaert, B. & Bruneel, H. (2004) Discrete-time queues with generally distributed service times and renewal-type server interruptions. *Perform. Eval.,* 55**,** 277-298.

Floyd, S. (2000) Recommendations On Using the Gentle Variant of RED. *http://www.aciri.org/ floyd/red/ gentle.html.*

Floyd, S., Gummadi, R. & ShenkeR, S. (2001) Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. *AT&T Center for Internet Research at ICSI*.

Floyd, S. & Jacobson, V. (1993) Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.,* 1**,** 397-413.

G.Thiruchelvi & J.RAJA (2008) A Survey On Active Queue Management Mechanisms *IJCSNS International Journal of Computer Science and Network Security,* 8.

Hamadneh, N., Murray, D., Dixon, M. & Cole, P. (2012) Dynamic Weight Parameter for the Random Early Detection (RED) in TCP Networks. *International Journal on New Computer Architectures and Their Applications (IJNCAA),* 2**,** 342-352.

Kiruthiga, B. & RAJ, E. G. D. P. (2014) APAQM: An Alternate Path AQM Congestion Control Algorithm. *International Journal of Computer Science and Mobile Computing,* 3 93-99.

Kusumawardani, M. (2013) Active queue management (aqm) and adaptive neuro fuzzy inference system (anfis) as intranet traffic Control. *Academic Research International* 4.

Lapsley, D. & Low, S. (1999) Random early marking: an optimisation approach to Internet congestion control. *Networks, 1999. (ICON '99) Proceedings. IEEE International Conference on.*

Lin, D. & Morris, R. (1997) Dynamics of random early detection. *ACM SIGCOMM.* New York.

Madipelli, S., Gillella, D. R. & Devaraya, S. (2009) The RED Algorithm – Averaged Queue Weight Modeling for Non Linear Traffic. *Science in Electrical Engineering with Emphasis in Telecommunication.* Blekinge Institute of Technology.

Neely, M. J., Modiano, E. & Chih-PinG, L. (2008) Fairness and Optimal Stochastic Control for Heterogeneous Networks. *Networking, IEEE/ACM Transactions on,* 16**,** 396-409.

Ott, T. J., Lakshman, T. V. & Wong, L. (1999) SRED: stabilized RED. *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.*

Perros, H. G. & Elsayed, K. M. (1994) Call Admission Control Schemes: A Review. North Carolina State University at Raleigh.

Ramakrishnan, K. K. & RAJ, J. (1988) A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer. *Symposium proceedings on Communications architectures and protocols.* Stanford, California, USA, ACM.

Salim, J. H. & Ahmed, U. (2000) *Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks*, RFC Editor.

Singh, S. & Balveer, S. (2013) Performance & Analysis of REM, RED, & GREEN AQM Algorithms in Congestion Control. *International Journal of Advanced Research in Computer Science and Software Engineering,* 3.

Tanenbaum, A. S. (2002) *Computer Networks*, Prentice Hall Ptr.

Thiruchelvi, G. & Raja, J. (2008) A Survey On Active Queue Management Mechanisms. *IJCSNS International Journal of Computer Science and Network Security,* 8**,** 130-145.

WELZL, M. (2005) *Network Congestion Control: Managing Internet Traffic,* Chichester.

Woodward, M. E. (1993) *Communication and Computer Networks: Modelling with discrete-time queues*, Wiley-IEEE Computer Society Press.

Yaghmaee, M. H. & Amintoosi, H. (2003) A Fuzzy Based Active Queue Management Algorithm. *Computer Department, Ferdowsi University of Mashhad, Faculty of Engineering, Mashad***,** 458-462.