ijsae Online Journal

*International Journal Of Scientific Research And Education*
||Volume||2||Issue|| 7||Pages 1364-1371||July-2014|| ISSN (e): 2321-7545
Website: http://ijsae.in

# EAGRED: A EnhanceVersion of Active Queue Management Algorithms of Congestion Avoidance

Authors

***[1]Rajnish Kumar Chaturvedi,[2]Jyoti Kesarwani***

[1]M.Tech.Scholar Department of Computer Science & Engineering, United Collage of Engineering & Research, Allahabad, Uttar Pradesh, INDIA.

[2]. Assistant Professor Department of Computer Science & Engineering, United Collage of Engineering & Research, Allahabad, Uttar Pradesh, INDIA

Email-[1]chaturvedi040@gmail.com

## ABSTRACT

*Problem of Quality of Service (QoS) providing in computer networks in case of uncertainty of data transferring processes is examined. Method of random early detection (RED), gentlerandom early detection (GRED) and adoptive gentle random early detection (AGRED) for congestion avoidance in networks is analyzed. Modification of enhance adaptive gentle RED (EAGRED) is offered. This modification consists in adjusting parameter, that are used during determination of average queue length in router's buffer, on a base of queue stability condition and dynamic adjustment of maximum probability of packets drop in case of congestions. Reducing of networks parameters amount, that are used for adjusting of active queue management algorithms, on a base of their ranking towards availability and importance, is offered.*

***Keywords***—*Quality of Service; congestion avoidance;adaptive control*

## I. INTRODUCTION

Practical problem of Quality of Service (QoS)providing for large amounts of heterogeneous traffic hasspecial importance in modern telecommunicationnetworks in case of dynamics uncertainty of networkflows and network environment during datatransferring.As known, quality of service providing while datatransferring consists in supporting such parameters asavailable bandwidth, priority, data transmission delays,delay variation (jitter), packet losses at a level thatcorresponds to a particular class of service.

QoS providing is important for adequate functioningof computer information networks that are evolvingtowards implementing a concept of next generation networks and wireless Ad Hoc networks.Control methods of network resources and datatransferring play special role among existing QoS tools.Adaptive network control is one of existing ways for QoSproviding in case of parametric uncertainty of datatransferring processes and network environment.

Control system of data transferring should be able toadapt to a variable network behavior. It's different to thetraditional approach that provides a control of data flowson a base of predefined rules of service. But in realconditions control problem is complicated byheterogeneity of transferring data in real time anduncertainty of network environment.

QoS improving in computer networks and automaticconfiguration of algorithms for network traffic controlaccording to changes of users' amount, their needs andrequirements for quality of services, equipment,communicationlinks, is actual task.

The goal of this paper is to develop a method of Enhance adaptive control of transferring data flows inheterogeneous networks for Quality of Service providingfor real-time traffic.

## II. RANDOM EARLY DETECTION ALGORITHM

Almost 90% of the traffic in modern networks istransmitted by TCP (Transmission Control Protocol). Therefore, let's consider a control of data transferringprocesses in computer networks on a base of TCP/IP (Internet Protocol) stack.Data volume can greatly exceed the abilities of itstransferring due to features of traffic. Overloading occursin case of insufficiency or nonconformance of networkresources to current load and in case of inefficientdistribution of data flows to available resources.

Methods of congestion avoidance are based onnetwork traffic monitoring and provide an opportunity tocongestion prediction in network bottlenecks and preventits occurrence. Congestion control methods are aimed toremove already appeared congestions.

Algorithms of Random Early Detection (RED) areused in TCP/IP networks. RED consists in packetsrandom dropping during load increasing. Problem ofqueue tail dropping appears in case of high priority traffic.Queue overloading must be examined to provide theintegrity of such traffic. Packets with lower priority mustbe dropped on a base of some criterion before packetswith higher priority will be dropped.

RED algorithm is one of the most common controlalgorithms for network queues in TCP/IP networks.

## RED:-

Random Early Detection algorithm detects incipient congestion at router buffer in preliminary stages. RED consists of two separate algorithms. The first algorithm is for calculating the average queue length (aql) which is calculated based on the following formula:

$$aql = aql \times (l - qw) + qw * queueSize \qquad (1)$$

whereqw is queue weight and queue Size is instantaneous queue length.

The second algorithm is for calculating the packet marking probability which defines how frequently the gateway marks the packets at given current level of congestion(S. Floyd & Jacobson, 1993). The formula for calculating the dropping probability is shown as follow:

$$Dp = Dinit / (1 - C * Dinit) \qquad (2)$$

Here C is a counter that represents the number of packets arrived at router buffer and has not dropped since the last packet was dropped and Dinit which is the initial packet dropping probability is defined as follow:

$$Dinit = Dmax+[(1 - Dmax) * (aql - max.\ threshold) / max.threshol\ ] \qquad (3)$$

RED decides to drop a packet with comparison of calculated aql with two thresholds which are min threshold and max threshold as represented in figure 1. If the aql is smaller than min threshold no packet will be dropped and no congestion has occurred yet and it continues to add packets to the queue. if the aql is bigger than max threshold, a sever congestion occurred and every arriving packet will be dropped with Dp = 1. And if aql is between the min threshold and max threshold RED needs to calculate the Dp in order to add or drop packet.

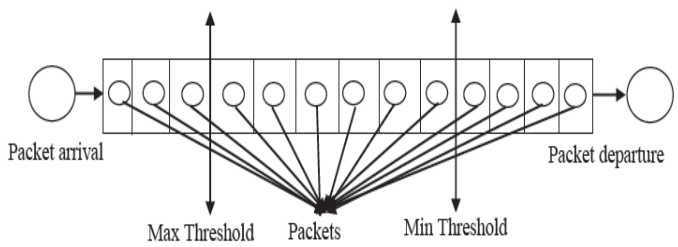


**Figure 1.** Single router buffer for RED

## GRED:-

GRED process is almost same as RED but the man difference is in parameter setting in order to be optimized and have a better performance regarding to Packet loss and throughput. In GRED another parameter was introduced namely Double max threshold which is illustrated in figure 2.

## AGRED:-

As mentioned in introduction AGRED tries to overcome high packet loss issue in GRED. This issue was settled with modification in calculation of initid dropping probability (Dinit) in the situation ifml is between max threshold and Double max threshold which is presented in Eq. (4).

$$Di\ nit = Dmax + \{ [(1 - Dmax) / 2 ]* (ml - max.\ threshold) / mw.threshol \} \qquad (3) \qquad (4)$$

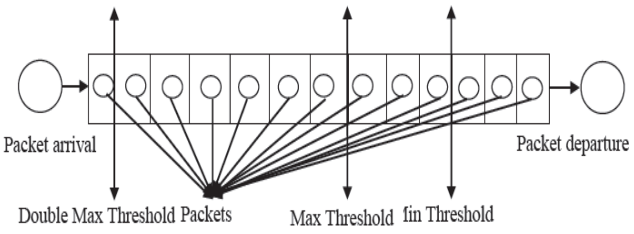


**Figure 2.** Single router buffer for GRED and AGRED

## ENHANCED ADOPTIVE GANTLE RANDOM EARLY DETECTION ALGORITHM (EAGRED)

Enhance adaptive gentle RED (EAGRED) is asubclass of Random EarlyDetection algorithm and has been offered in papers. EAGRED stabilizes an average queue length relatively toa predetermined target size of a queue.

EAGRED uses following target range to provide highbandwidth and low delays during data transferring

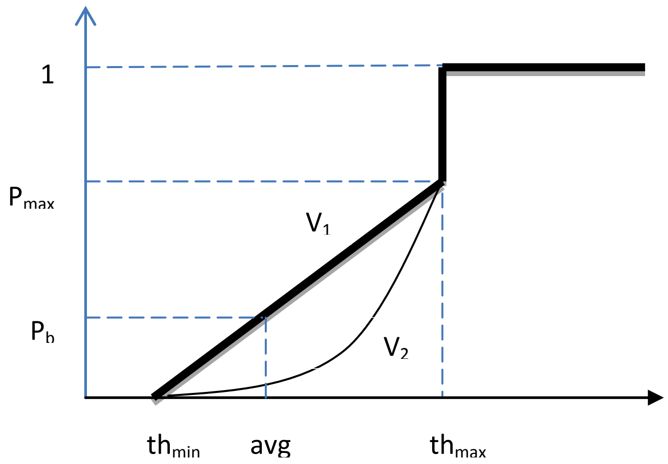


**Figure 3:** Two different function to evaluate $P_b$

The curve $V_1$ represents the drop probability of RED algorithm that is linear curve and the curve $V_2$ represents the enhanced adoptive gentle RED algorithm that is the exponential curve.

RED calculates the probability of dropping a packet by 2 steps when average queue size falls in between th_min and th_max. The first step is to calculate a probability called Pb, which is based on the formula below,

$$Pb = Pmax*(avg - th\_min)/(th\_max-th\_min) \qquad (4)$$

It can be seen that the probability Pb is calculated based on a linear function.

In the second step, RED counts the number of packets have gone through the gateway (count) since last packet is dropped from the flow and apply the following formula,

$$Pa = Pb / (1 - count*Pb) \qquad (5)$$

We calculate the drop probability of EAGRED algorithm by modifying the equations (4). we can express the formula for function v1 the convex function (we chose exponential function) using th_max, v_b and v_a as below,

$$Pb = Pmax*exp(lm) \qquad (6)$$

where,

$$lm = (v\_ave - th\_max) / Xmin \quad (7)$$

*and*,

$$Xmin = -(v\_b/v\_a) \qquad (8)$$

## EVALUATION RESULTS

The performance of the proposed EAGRED algorithm is com- pared with those of GRED, AGRED, and RED. The performances of these algorithms are measured ten times in ten runs, each taking different seeds as input to the random number generator. This step removes possible bias in the output results and producescondense intervals for the performance measures. The performances of all AQM methods are calculated after the system reaches a steady state. For the parameter settings, RED, GRED, and AGRED are initiated using identicalparameters at most. To create congestion and non-congestion scenarios at the buffer, the probability of packet arrival was set to several values; each value tends to create a congestion or non-congestion status. The buffer size room of 20 packets was used to detect congestion at small buffer sizes. The total number of slots used in the experiments was 2000000. This performance measures and encapsulates a sufficient warm-up period. The warm-up period is terminated when the system reaches a steady state. The *minthreshold*, *maxthreshold*, *D*max, and *qw* values are set to 3, 9, 0.1 and 0.002, respectively, as recommended in RED. Finally, the *doublemaxthreshold* value is set to 18 as recommended in GRED. Table 1 lists all the utilized parameters. The simulation results are measured using several performance metrics (e.g., *mql*, *T*, *D*, *PL*, and *Dp*), which are discussed in the following subsection. value allows the incorporation of accurate

**Table 1. Parameter settings for GRED, AGRED and RED algorithms**

| Parameter | EAGRED | RED,GRED,AGRED |
|---|---|---|
| Probability of packet arrival | 0.18-0.93 | 0.18-0.93 |
| Probability of packet departure | 0.5 | 0.5 |
| Router buffer capacity | 100 | 100 |
| Qw | 0.002 | 0.002 |
| Dmax | 0.1 | 0.1 |
| Number of slots | 2000000 | 2000000 |
| Minthreshold | 20 | 20 |
| maxthreshold | 60 | 60 |
| doublemaxthreshold | ........... | ------------- |

Mql, throughput, and delay:-

Figures 4-6 illustrate the output performancesof RED, GRED, AGRED, and AEGRED using different probabilities of packet arrivals.Specically, Figure 4 illustrates the *mql* versus the probability of packet arrival.

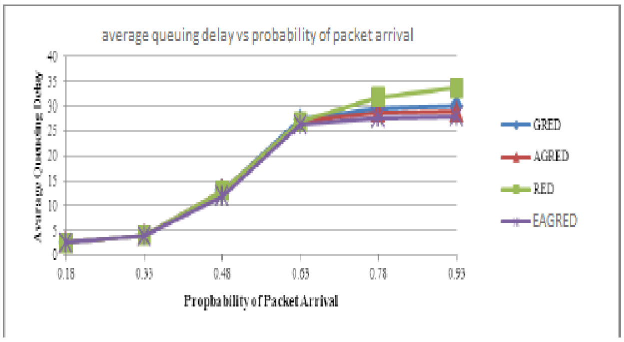


**Figure 4**.*mql*vs. probability of packet arrival

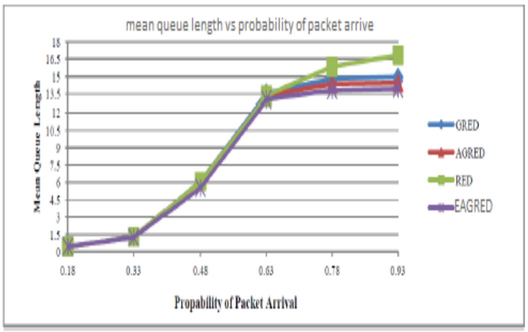


**Figure 5.***D* vs. probability of packet arrival

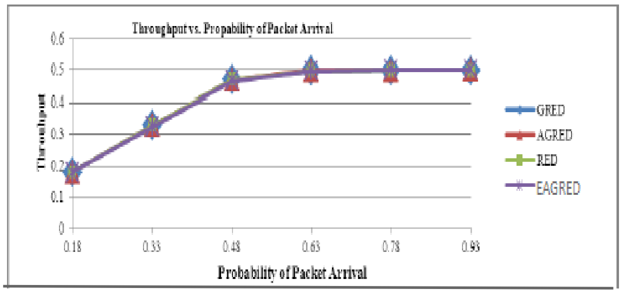


**Figure 6**.*T* vs. probability of packet arrival

**Packet loss and *Dp***

The proposed EAGRED algorithm is compared with the RED, GRED, and AGRED algorithms in terms of *PL* and *DP* in this subsection. The goal of the conducted comparison is to show the quantity of packets dropping at the router buffer in all compared algorithms. The performance measure results of *PL* and *DP* are computed after the system reaches a steady state. The results of *PL* and *DP* are obtained as before by running the algorithm simulations ten times with various random seeds, then taking the mean of the ten results. The performances of RED,GRED, ARED, and EAGREDalgorithms in terms of *PL* and *DP* are illustrated in Figures 7 and 8, respectively.

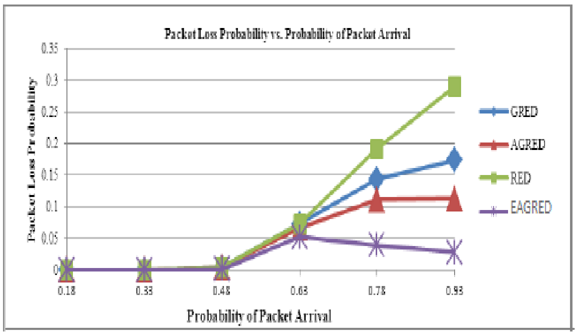


**Figure 7**.*PL* vs. probability of packet arrival

## CONCLUSIONS AND FUTUREWORK

A comparison has been conducted between themethods of EAGRED, RED and the Adaptive GRED with reference to several performance measures, i.e. Q_*avg*, $P_b$, *throughput etc* . This comparison aimed to identify which method offers more satisfactory performance measures. A decision which method offers more satisfactory performance measure results is only made depending on varying the parameter.

As part of on-going work, several extensions to EAGRED are being considered. In particular, additionalmechanisms for managing non-responsive flows are being examined. In this paper, non-responsive flowswere rate-limited to a fixed amount of bandwidth across the bottleneck link. However, it is possible torate-limit non-responsive flows to a fair share of the link's capacity. One way to do this is to estimate boththe number of non-responsive flows and the total number of flows going through the bottleneck. Using thisinformation, the rate-limiting mechanism can be set accordingly. Finally EAGRED queue management algorithm which is similarto "enhanced" RED is being considered.

## RRFERENCES


[1] G. Thiruchelvi and J. Raja, A survey on active queue management mechanisms, *IJCSNS Interna- tional Journal of Computer Science and Network Security*, vol.8, 2008.

[2] M. Welzl, Network congestion control, *Proc. of Managing Internet Traffic*, Chichester, UK, 2005.

[3] A. S. Tanenbaum, *Computer Networks*, 4th Edition, Prentice Hall Ptr, 2002.

[4] D. Lin and R. Morris, Dynamics of random early detection, *Proc. of ACM SIGCOMM*, New York, NY, USA, pp.127-137, 1997.

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

[6] S. Floyd, *Recommendations on Using the Gentle Variant of RED*, http://www.aciri.org/oyd /red/gentle.html, 2000.

[7] M. Baklizi et al., Performance assessment of AGRED, RED and GRED congestion control algo- rithms, *Information Technology Journal*, vol.11, pp.255-261, 2012.

[8] J. Ababneh et al., Derivation of three queue nodes discrete-time analytical model based on DRED algorithm, *Proc. of the 7th International Conference on Information Technology: New Generations*, pp.885-890, 2010.

[9] H. Abdel-jaber et al., Traffic management for the gentle random early detection using discrete- time queueing, *Proc. of International Business Information Management Conference*, Marrakech, Morocco, pp.289-298, 2008.

[10] H. Abdel-Jaber et al., Performance evaluation for DRED discrete-time queueing network analytical model, *Journal of Network and Computer Applications*, vol.31, pp.750-770, 2008.

[11] A. Moare_anpour and V. J. Majd, Input-to-state stability in congestion control problem of computer networks with nonlinear links, *International Journal of Innovative Computing, Information and Control*, vol.5, no.8, pp.2091-2106, 2009.