

AN EVENT DRIVEN SIMULATION FOR ADAPTIVE GENTLE RANDOM EARLY DETECTION (AGRED) ALGORITHM

Omid Seifaddini, Azizol Abdullah and Shamala Subramaniam

Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia, Serdang, Malaysia

Received 2014-02-25; Revised 2014-02-27; Accepted 2014-05-13

ABSTRACT

Simulations are used to find optimum answers for problems in wide areas. Active queue management algorithms such as RED, GRED, typically use simulators like ns2 which is an open source simulator or OPNET, OMNET which are commercial simulators. However, beside the benefits of using simulators like having defined modules, parameters. There are problems such as complexity, large integrated components and licensing cost. To have an ideal balance in mentioned benefits and problems and to further complement the repository of simulators, this study presents the description of a general-purpose programming language based discrete event simulation for active queue management. This research has focused at developing a discrete event simulator to implement one of active queue management algorithms which is called AGRED. The results showed that the developed simulator has successfully produced the same results with an average deviation of 1.5% as previous simulator in AGRED.

Keywords: Discrete Event Simulation, Simulation, AGRED, Congestion Control, Queue Management

1. INTRODUCTION

Based on the (Abdel-Jaber *et al.*, 2012; Mahmoud Baklizi and Abdel-Jaber, 2012; Reddy and Ahammed, 2008) and (Floyd and Jacobson, 1993) congestion is fundamental issue in computer networks. Congestion happens at the router buffer when incoming packets exceeds available network resources such as buffer space. Congestion leads to increase packet loss, queuing delay and finally degrades the throughput.

Different Active Queue Management (AQM) algorithms deal with the congestion in different ways such as Random Early Detection algorithm (RED) which is one of the most well known active queue management algorithms and was recommended by Internet Engineering Task Force IETF (Braden, 1988). RED was proposed in (Floyd and Jacobson, 1993) in which a version of REAL simulator built on Columbia's Nest simulation package with lots of modifications was used to evaluate the proposed algorithm.

Although RED was proposed to deal with the congestion but there were issues such as parameter setting which were stated in (Mahmoud Baklizi and Abdel-Jaber, 2012). Gentle Random Early Detection (GRED) was proposed (Floyd, 2000) to solve the primarily issues with RED. GRED was evaluated using the same simulation as used in RED.

Even though the GRED proposed to overcome the limitation of the RED but the packet loss rate is still high in congestion situation. An Adaptive Gentle Radom Early Detection (AGRED) was proposed by (Mahmoud Baklizi and Abdel-Jaber, 2012) to derive better quality result in packet loss. AGRED simulation was implemented using java programming language.

There exists several types of simulators, each associated with drawbacks such as large integrated components, licensing cost, complexity and being generalized such that if someone intend to use the simulator, lots of modification is needed (Floyd and Jacobson, 1993). Beside that AGRED was simulated using java programming languages but the details were not given in the study.

Corresponding Author: Omid Seifaddini, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Malaysia

This study presents the details of development of General Purpose Programming Languages (GPL) based Discrete Event Simulation (DES) to do experimental part of selected paper in a way that it is small, specified, light and free of license.

The rest of the study is presented in the following order in section 2, an Analysis tool is proposed along with its components. In section 3 the results of proposed simulation which is being tested on AGRED algorithm is discussed and finally in section 4 conclusion and future work are presented.

1.1. Proposed Analysis Tool

This section is about the deliberation of the chosen algorithm named Adaptive Random Early Detection (AGRED). The description of this algorithm is given form the core to provide the validation of the developed simulator. The proposed and developed simulator needs to be verified and validate and is highly critical and mandatory in serving its acceptance. The comparative simulator, from (Mahmoud Baklizi and Abdel-Jaber, 2012), shall be referred as a benchmark hereafter. The benchmark attributes encompass the purpose of creating comparison of this research within the same domain with another method which is distinct in the contexts of its mechanics and platform (Naghshbandi *et al.*, 2012).

1.2. AGRED

AGRED is one of the Active queue management algorithms to detect incipient congestion at the router buffers. AGRED algorithm consists of two separate algorithms same as RED algorithm (El Emary and Taweel, 2005). The first algorithm is for calculating the Average Queue Length (AQL) and it determines the degree of burstiness that will be allowed in the gateway queue. The formula for calculating average queue length is given as follow Equation 1:

$$aql = aql \times (1 - q_w) + q_w * queueSize \quad (1)$$

where, q_w is queue weight and $queueSize$ is the instantaneous queue length.

The second algorithm which is for calculating the packet marking probability determines how frequently the gateway marks packets, given the current level of congestion. The goal is for the gateway to mark packets at fairly evenly spaced intervals, in order to avoid biases and avoid global synchronization and to mark packets sufficiently frequently to control the average queue length

(Floyd and Jacobson, 1993). The formula for calculating the dropping probability is given as follow Equation 2:

$$Dp = D_{init} / (1 - c * D_{init}) \quad (2)$$

where, 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 D_{init} is defined as follow Equation 3:

$$D_{init} = D_{max} + [(1 - D_{max}) * (aql - max\ threshold) / max\ threshold] \quad (3)$$

The process of the AGRED algorithm is presented in **Fig. 1** and AGRED model is shown in **Fig. 2**.

Figure 1 illustrates the flowchart of how AGRED decides to whether drop packets from queue or to add packets into the queue. The algorithm will compute the aql for every packet arrival and then it compares the aql with the three thresholds presented in **Fig. 2**. It continues to add packets into the queue if the aql is less than $min.threshold$ and it drops all arriving packets if aql is bigger than double max threshold. Otherwise AGRED needs to calculate the dropping probability to decide whether to add or to drop arriving packets if the aql is between min and $max.threshold$ or $max.threshold$ and double max threshold.

1.3. Simulation Environment

The AGRED simulator was implemented using general purpose programming language, Java. For packet generation purpose, geometrical distribution is used as it used in the benchmark paper. The parameter used in this simulation is given in **Table 1**. The experimental results of proposed simulation environment are carried out by using these values in order to validate proposed simulator with (Mahmoud Baklizi and Abdel-Jaber, 2012).

1.4. Performance Metrics

In computer terms performance metrics are measures used to evaluate and assess given algorithm, model, framework such as throughput, delays. The performance metric has been used in this study to evaluate the simulator is Packet Loss (PL). Packet loss happens when one or more packets fail to reach destination while travelling across computer networks. In this study packet loss is defined as deduction of number of departed packets (ndp) from number of arrived packets (nap) as presented in Equation 4:

$$Pl = nap - ndp \quad (4)$$

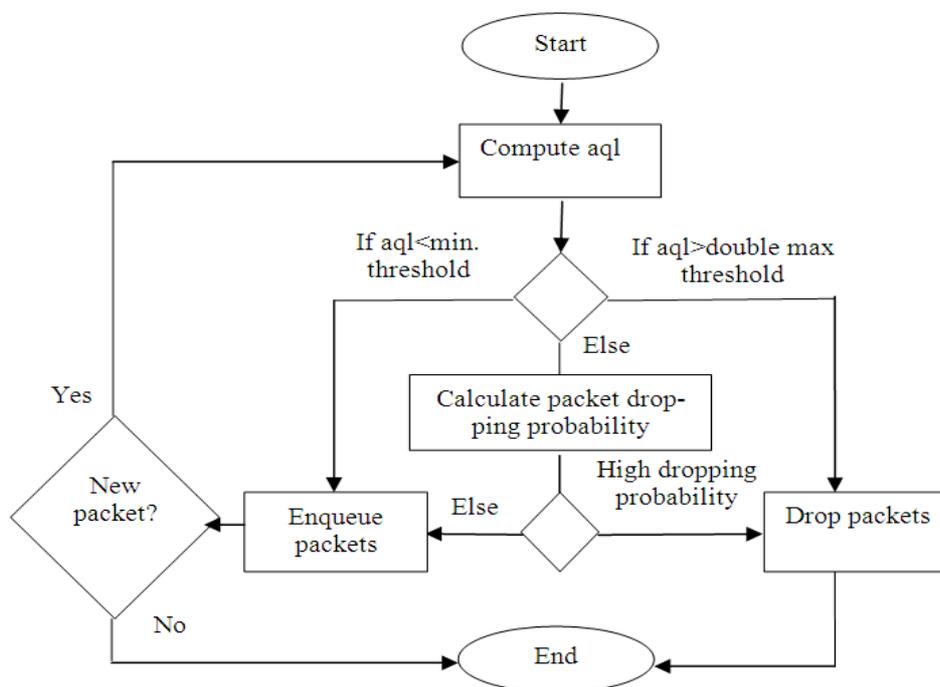


Fig. 1. AGRED Flowchart

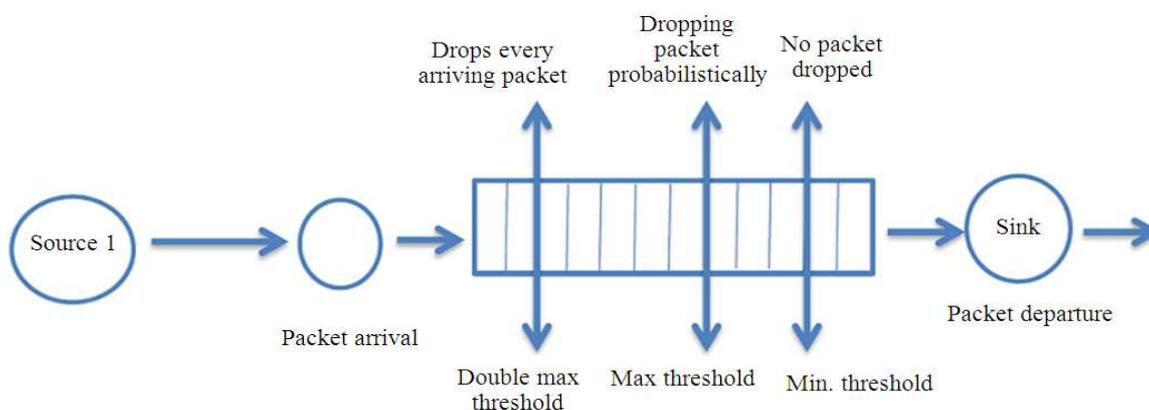


Fig. 2. The single router buffer with AGRED

1.5. Simulation Structure

In this section the derivation of simulator is explained. For writing a simulator the events need to be derived and for this purpose understanding of what is the statistical composition is pre requested.

Statistical composition of a system is one singular representation that determines the events (Fuaad *et al.*, 2009). In this study the statistical composition is the number of packets. In discrete event simulations, a

chronological sequence of events represents the operation of the system (Tanha *et al.*, 2012). An event is an occurrence that changes the statistical composition of a system. Since the statistical composition is defined as the number of packets, what increases or decreases the number of packets. In the Fig. 3, Deliberation of AGRED algorithm is depicted to illustrate the events that affect the number of packets.

The following steps of AGRED algorithm are derived as events regarding Fig. 3:

1) System starts at idle state (Number of Packet=0, Buffer Size=0, Sink status= Idle)
2) The source generates packets. [Arrival]
3) The source transmits the packets to the sink.
4) The status of the sink is checked:
4.1) if the sink is idle admit the packet into the sink to be processed.
4.2) if the sink is busy/ occupied admit the packet into the router buffer due to buffer availability. (Calculate the AQL (Average Queue Length) for the arriving packet).
4.2.1) If $AQL \geq \text{Double max threshold}$, mark / drop arriving packet with $Dp = 1$.
4.2.2) Check if $\text{Min. threshold} \leq AQL < \text{Double max threshold}$, mark/drop arriving packet probabilistically in terms of its Dp value.
4.2.3) Check if $AQL < \text{Min. threshold}$ (the queue is available), enqueue the packet into buffer (no packet have dropped).
5) The packet in the sink which has completed of being processed will leave the system. [Departure]
5.1) check the buffer size; if $\text{Buffer Size} > 0$ select a packet from the router buffer and place in the sink in order to be processed.
5.2) if the queue is empty then make the sink idle.

Fig. 3. AGRED deliberation

Step 2: The source generates packets: the number of the packets is increased. Therefore, it is considered as an event (packet arrival)

Step 5: Leaving of the packets from the system is considered as an event due to decrease in number of the packets (packet departure)

For packet generation purpose, geometrical distribution is used as used in AGRED (Mahmoud Baklizi and Abdel-Jaber, 2012). In the simulator, geometrical distribution represents the types of generated packets which

are text in reality. The pseudo code of geometrical distribution used in our simulator is presented in Fig. 4.

Scheduler is part of the simulator and is a component which picks the event/function based on its execution time. In the other words, the scheduler is not in the charge of assigning time to the events. The only task of this component is choosing from the list of the events. In this project the scheduler task is to pick between the arrival and departure function based on the smallest time. The mentioned deliberation of algorithm, steps and scheduler are shown in detail in Fig. 5.

```

public double geo(double mean) {
    Random r = new Random();
    double ran = r.nextDouble();
    return Math.ceil(Math.log(ran) / Math.log(1.0 - (1/mean)));
}
    
```

Fig. 4. Geometrical distribution pseudo code

```

nextDeparture:=1.0e+30;
nextArrival:=0;
simulationClock:=0;
packetsInQueue:=0;
endTime:= 2000000;
while((nextArrival < endTime) || (nextDeparture < endTime) ) do
    begin
        if(nextArrival < nextDeparture) then
            simulationClock:=nextArrival;
            nextArrival:=simulationClock+arrivalTime;
            if(packetsInQueue <= 1) then
                nextDeparture:= simulationClock+serviceTime;
                packetCouldTransmitted:= (128 * (simulationClock-idleTime));
                aql:= Math.pow((1-qw), packetCouldTransmitted) * aql;
            else
                aql:=(1-qw)* aql + (qw*queueSize);
            if(aql < minThreshold) then
                C:=-1;
                queue.add();//add one packet to the queue
                packetsInQueue:= packetsInQueue+1;
                D:=0;
            elseif (aql >= mixThreshold && aql <maxThreshold) then
                C:=C+1;
                Dinit:=(Dmax*(aql-minThreshold))/(maxThreshold-minThreshold);
                Dp:=Dinit/(1-C*Dinit);
                C:=0;
                if(Dp < Dmax) then //adding packet to the queue
                    queue.add();
                    packetsInQueue:= packetsInQueue+1;
                else // dropping packets probabilistic
                    queue.remove(); //remove a packet from queue
                    packetsInQueue:= packetsInQueue-1;
            elseif (aql >= maxThreshold && aql < doubleMaxThreshold) then
                C:=C+1;
                Dinit:=Dmax+(((1-Dmax)/2)*(aql-maxThreshold))/maxThreshold;
                Dp:=Dinit/(1-C*Dinit);
                if(Dp < Dmax) then
                    queue.add();
                    packetsInQueue:= packetsInQueue+1;
                else
                    queue.remove();
                    packetsInQueue:= packetsInQueue-1;
                C:=0;
            elseif(aql >= doubleMaxThreshold) then
                C=0;
                queue.remove();
                packetsInQueue:= packetsInQueue-1;
            else
                simulationClock:=nextDeparture;
                queue.remove();
                if(packetsInQueue >0) then
                    nextDeparture:=clock+serviceTime;
                else
                    idleTime:=clock; //starting time of idle period
                    nextDeparture=1.0e+30;
            endWhile
    
```

Fig. 5. AGRED simulator pseudo code

2. RESULTS

This section evaluates and validates the simulator by several mean inter arrival times and a mean service time which are given in **Table 1**. These parameters are based on the study (Mahmoud Baklizi and Abdel-Jaber, 2012) and they are intended to create non congestion and congestion situations. The objective of the evaluation is to verify the ability of the developed AGRED simulator to detect instantaneous congestion at router buffer in preliminary stages.

Figure 6 represents the results of simulation that is packet loss rate over different packet arrival rate. Packet loss rate is calculated based on the equation (4) in section 2.3. In **Fig. 6** packet loss means percentage of packets that fail to reach destination from total number of packets. For example when packet arrival probability (α) is 0.78, the packet loss is 0.08 which means 8% of total packets failed to reach the destination.

3. DISCUSSION

The two graphs in **Fig. 6** present the benchmark and propose simulator. The collected results are compared

with the results from paper (Mahmoud Baklizi and Abdel-Jaber, 2012). When α is 0.18, 0.33 and 0.48 packet loss rate is 0 because congestion has not happened yet and all packets reach destination successfully. When α is increased to 0.63, 0.78 and 0.93, sever congestion happens and some packets fail to reach the destination. As **Fig. 6** illustrates that proposed simulator has produced almost the same results as the benchmark with only 1.5% difference. Using random number generator for packet generation could be one of the reasons that slight difference occurs. Thus the proposed simulator was successfully able to simulate the behavior of packet loss parameter.

3.1. Limitations

This study focuses only on simulating the behavior of AGRED algorithm with the same parameters with benchmark study (Mahmoud Baklizi and Abdel-Jaber, 2012).

This simulation shows the behavior of AGRED algorithm for text traffic only and another types of traffic like audio and video are left for future investigation.

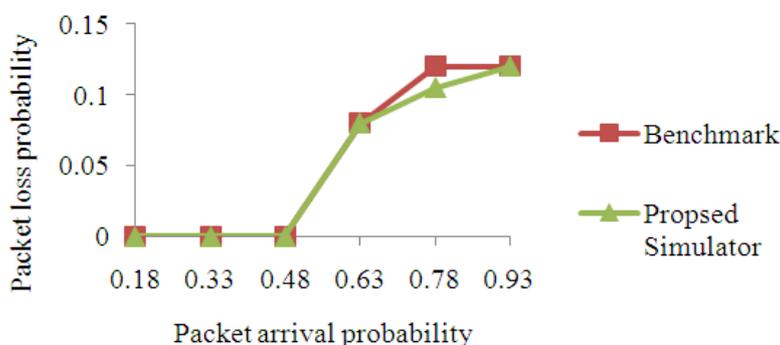


Fig. 6. Packet loss rate Vs packet arrival probability

Table1. Simulation parameter setting

	Parameter	Value
a.	Queue size	20
b.	Simulation time (end time)	2,000,000
c.	Mean inter arrival time (α)	0.18, 0.33, 0.48, 0.63, 0.78, 0.93
d.	Mean service time	0.5
e..	Min threshold	3
f.	Max threshold	9
g.	Double max threshold	18
h.	The maximum value of Dropping (D_{max})	0.1
i.	Queue weight	0.002

4. CONCLUSION

This study has significantly developed a General-Purpose Programming Language (GPL) Discrete Event Simulation (DES) for the analysis of congestion control algorithms. This specified simulator provided an easy simulation configuration for AGRED. The structure of presented simulation makes it very flexible in order to find optimum value for the parameters and future usage. To verify and validate the developed simulator, various mean inter arrival time were examined. The results showed that our simulator has produced the benchmark results in congestion and non congestion situations and successfully fulfills the requirements.

As mentioned in the section 5, the simulator implements only AGRED with geometrical distribution but also, the other algorithms such as GRED and RED and other distribution such as exponential and ON-OFF model can be implemented using the same simulators with the slight modification.

5. REFERENCES

- Abdel-Jaber, H., F. Thabtah, A.M. Daoud and J. Ababneh, 2012. Performance investigations of some active queue management techniques using simulation. *Int. J. New. Comput. Archit. Appl.*, 2: 286-301.
- Braden, B., 1988. RFC2309: Recommendation on queue management and congestion avoidance in the internet.
- El Emary, I.M.M. and F.A. Taweel, 2005. An advance approach to evaluate the performance of the TCP networks. *Am. J. Applied Sci.*, 2: 1375-1379.
- Floyd, S. and V. Jacobson, 1993. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1: 397-413. DOI: 10.1109/90.251892
- Floyd, S., 2000. Recommendations on using gentle variants of RED.
- Fuaad, H.A., S. Shamala, M. Othman and Z. Zuriati, 2009. A discrete event modeling and simulation of wave division multiplexing unidirectional slotted ring metropolitan area network. *J. Comput. Sci.*, 5: 456-465. DOI: 10.3844/jcssp.2009.456.465
- Mahmoud Baklizi, S.R. and H. Abdel-Jaber, 2012. Performance assessment of AGRED, RED and GRED congestion control algorithms. *Inf. Technol. J. Naghshbandi, D., S. Subramaniam and S. Rahimi Moghadam*, 2012. An event-driven simulation for Highest Urgency First (HUF): A latency and modulation aware bandwidth allocation algorithm for WiMAX base stations. *Wirel. Pers. Commun.*, 69: 39-52. DOI: 10.1007/s11277-012-0559-9
- Reddy, T.B. and A. Ahammed, 2008. Performance comparison of active queue management techniques. *J. Comput. Sci.*, 4: 1020-1023. DOI: 10.3844/jcssp.2008.1020.1023
- Tanha, M., S.D.S. Torshizi and S. Shamala, 2012. A discrete event simulator for extensive defense mechanism for denial of service attacks analysis. *Am. J. Applied Sci.*, 9: 909-916. DOI: 10.3844/ajassp.2012.909.916