Comparative Analysis of Three Active Queue Management (AQM) Techniques (RED, SFQ, REM) in Terms of Their Relative Effectiveness in Controlling Congestion in Networks

EGBA IFEANYICHUKWU[1], ALALIBO, TAMUNO-OMIE[2]
[1](B.Tech, M.Sc, IITP), River State College of Health Science and Technology, Port Harcourt, Rivers State. Nigeria
[2](B.Tech, M.Sc), Rivers State University, Port Harcourt, Rivers State. Nigeria.

***Abstract:* - There is now ever increasing demand for fast deployment of network infrastructures as well as for fast transfer of large quantity of data. But this high demand has been greeted with network traffic called congestion. This congestion reduces the performance of the network due to also increase in packet loss. In cause of reducing this rise in the rate of packet loss to enhance the network performance, active queue management (AQM) techniques has been deployed. This AQM are router based technique for detection of traffic early in a computer network. The main ideas behind AQM is to detect traffic in the network and to alert the sender to minimize its rate of sending packets, thereby minimizing the amount of packets sent into the network and so control the impending traffic.**

In this paper three AQM congestion control algorithm like RED, SFQ, REM, are considered for survey and analysis with their limitations, advantages and disadvantages, in terms of delay, throughput and queue length, etc. The aim of this survey is to analyze three AQM algorithms, to have a clear view the best amongst them and on how to develop a new AQM algorithm (hybrid) that will give a better result compared to these existing algorithms.

***Keywords:* Random Exponential Marking (REM), Stochastic Fair Queing (SFQ), Random Early Detection (RED), Throughput, Delay, Queue length , Loss rate.**

I. Introduction

A lot of applications running over the internet require moving bulky data quickly over a data network with a high-speed; It also needs links connecting network nodes that has high-bandwidth. And all applications require subjection to congestion control mechanism for the internet stability to be maintained. Generally, in recent times there are several techniques that are used mainly to transmit data or information through wireless networks which is beneficial to the host. These benefits include communication, low cost, fast transmission at limited time and so many others but when such network is overloaded, it can cause network congestion.

1.1 What is congestion?

Congestion simply means overcrowding of an item or people in a particular place and time. This also occurs in the internet especially the network. Network congestion can be likened to the heavy traffic that occurs in a city at a particular time, therefore congestion in network refers to the state of limited network services, where the excess data load or information carried by the node or link affects the service quality causing the prevention or reduction in connection, loss of data, delay caused by the network responding slowly and delayed throughput (the amount of data packet transmitted per unit time).

1.1.1 How does network congestion occurs?

Congestion generally occurs when there is accumulation of a particular thing which results to hindrance and blockage, so also in network. Network congestion occurs when the routers forward more data to the network leading to probably an overflow in the network causing loss or delay of data packet. Nevertheless, because of this delay (traffic) or loss, the data is retransmitted increasing the congestion. In all of this we can simply say that congestion occurs when the transmitted data packet exceed the network holding capacity (Palanisamy, A., 2016).

1.1.2 Effects of congestion in a network

Congestion in a network has great impact on the productivity of a network, therefore, as network activities increases, so does the collision/ congestion rate which causes great defect on the performance rate of the network. The effect caused by data congestion includes

- Loss of data packet
- Network performance degradation
- Loss of network operators (host) because of delay
- Artificial congestion by the internet service provider
- Waste of resource utilization time

- Network throughput is reduced

1.1.3 Causes of network congestion

There are different causes of network congestion. They are:

1) Poor network design: The network design matters a lot when using the internet because the needed hardware or software are not used for the design and when used by the host causes slowdowns as a result of congested network.
2) Multicasting: This is a communication scheme, where several transmissions are transmitted from a source, but what makes it unique is that it is selective of the group it sends message to. Although multicast can also be a cause of network congestion in the sense that when two or more data packet are transmitted at the same time, as they move to their various destination there can sometimes be a collision between the packets causing the congestion in the network. An example is the message sent to e-mail's given.
3) Too many hosts in the domain: In such a situation congestion occurs because when so many phones or computer (host) is operating on a particular site (domain), this causes delay connection in the network. For example, registration for National Youth Service Program (NYSC).
4) Limited bandwidth: A bandwidth is the maximum information transmitted through a network in a given time. If the bandwidth cannot accommodate the transmitted information because it is low, this can also cause congestion in the network.
5) Screen storm: This deals with actions that affect the network where there are so many demands to be processed by the network and the network fails to execute all at the same time. An example is when there is promo on sales of different items.

For good network performance to be achieved, mechanisms must be provided to prevent the network from being congested for any significant period of time. This mechanism includes the congestion detection, the congestion control and the congestion avoidance. Therefore, there is need to detect the occurrence of the congestion in-order to determine how to control and avoid it.

1.1.4 Congestion Control

Congestion control is defined as a mechanisms and techniques that can either stop congestion before it tries to happen, or remove it after it has happened.

Congestion may occur in a network if the load on the network exceeds more than the network's capacity can carry. There is great need to control congestion in other to avoid collapses. Therefore, actions need to be taken both on the transmission protocols and the network routers in other to guarantee network stability, ensures fair resource allocation, throughput efficiency and maximize bandwidth utilization. This required action can only be done by a congestion control or avoidance protocol/mechanism.

2.0 AQM Algorithm/ Techniques

It is a congestion control algorithm that increases link utilization, throughput and decreases packet loss and delay. AQM as an algorithm senses in advance congestion that may occur in the network and alerts the sender to minimize its rate of sending packets so that the amount of packets flow in the network is reduced. There are several AQM algorithms but this paper will be considering only three of it , which are: RED, SFQ and REM

2.1 SFQ (Stochastic Fair Queuing)

It is one of the AQM algorithms which utilizes hashing and round robin algorithm. In this technique, four options are used to identify traffic flow: source address, source port, destination port and destination address. The options are used to classify the packet into sub-streams of 1024 by the SFQ hashing algorithm and then the robin algorithm is used to fairly distribute it to all the sub-stream. It can also be said to be a class of queue scheduling disciplines built to share a pretty large number of separate FIFO queues (P.E.1990). Fairness is achieved if the number of queues is to an extent Increased.

2.1.1 SFQ algorithm

Parameters

int maxqueue max queue size in packets

int buckets number of queues

2.1.2 SFQ Algorithm Functions

1. Enque the packet enque()
2. Dequeue the packet dequeue()
3. Calculate the hash function hash()
4. Calculate the fair share initsfq()

2.2 Random Early Detection (RED)

RED is the first known AQM algorithm. It can be called Random Early Detection, Random Early Drop or Random Early Discard which provides the technique needed to avoid congestion. It solves the Traditional drop tail algorithm problem which includes: global synchronization, dropping of packets when the buffer is filled and its inability to share among traffic flow buffer space fairly. RED solves this problem of drop tail through queue management ie by detecting in advance the incipient congestion and also communicating the same to the end-hosts, thereby allowing them to reduce their transmission rates before the queues begin to overflow that may cause the packets to start dropping. It watches the queue size based on the queue. And if the queue is empty, it accepts all the packets; it drops the incoming packets if the queue becomes filled.

RED activates its congestion detection mechanism by maintaining an exponentially weighted moving average of the queue length. In order to aid efficiency in RED, it must: (1) make sure that notification about congestion is sent at a speed enough to suppress the sources of transmission without having to underutilize the link. (2) It must also make sure the queue has a buffer space enough to store any added load that is greater than the link ability between when congestion is detected to when the added load diminishes at the bottleneck link while responding to that congestion notification.

2.2.1 Objectives of RED

The major RED algorithm objectives are:

- Monitor of the length of the queue
- utilization of high link
- Early detection of congestion
- Queuing delay minimization
- Reduction of loss of packet
- Achievement of fairness
- Avoidance of global synchronization

2.2.2 Random Early Detection (RED) Algorithm

Step 1: Calculate the average queue size AvgQueSize

Step 2: If (AvgQueSize > MaxQueSize)

drop the packet

Step 3: else if(AvqQueSize ==0)

queue empty

accepts all incoming packets

Step 4: else if(AvgQueSize=minqueuethreshold)

calculate dropping probability pa

drop the packet with probability pa

Step 5: else forward the packet.

2.3 Random Exponential Marking (REM)

It's an active queue management algorithm that is aimed at achieving both very high utilization and also negligible delay and loss in a clear and adaptable way in a versatile manner. It separates the congestion measures with its relative performance measure which includes delay, loss of throughput packet, the ratio Packet are delivered, etc and balances the measure in performance around the target unreservedly of the amount of clients. While congestion measure shows excess request for data transfer capacity and must also track the amount of clients, performance measure no-matter the amount of clients ought to be balanced out around their set targets. The idea behind REM technique are firstly, it endeavors to customer rates to the capacity of the network while clearing bolsters, independent of amount of users and secondly, it embeds the amount of congestion measures, added-up over every routers that connects the clients to the end-to-end dropping probability.

2.3.1 REM Functions

- Function Reset.
- To mark probability and compute average input rate price.
- To Enqueue
- To Dequeue

2.3.2 REM Algorithm

Parameters

double v_pl link price

double v_prob packet marking probability

double v_in input rate

double qib queue in bytes.

double remp_.p_gama value of gamma

double remp_.p_phi value of phi

double remp_.p_pktsize mean packet size

double remp_.p_updtime update time

double remv_.v_prob dropping probability

double curq current queue size

int pmark number of packets being marked.

3.0 METRICS of PERFORMANCE

Drop tail and RED are just the two only active queue management algorithm in the queue object of the hierarchy as completely shown in Figure 1;Therefore it is only under this object that the other discussed algorithm in the network congestion control section are implemented. All these other algorithms are affixed as a child to this element in the chain of hierarchy and the Queue operates their parent. Others can similarly also be sent into the **ns2** architecture and also make them run (http:www.isi.edu/nsam/ns/index.html).

3.1 Trace File Analysis

The Figure 1 and the Figure 2 shows the Class hierarchy chain of ns2 with depiction of every field showing up sample by an example of trace file and also their names. Right when the ns is run, the trace file can store the trace of every event. Tracing into an output ASCII file, the trace is actually as shown in figure 1 and 2 below composed into 12 fields.

Figure 1. The Class hierarchy of NS2

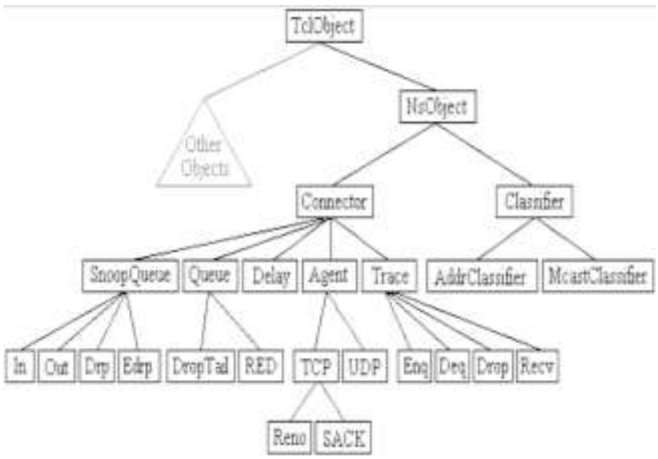


Fig. 2 Trace files structure

event	time	From node	To node	pkt type	prk size	flags	Fid	src addr	dst addr	Seq num	Prk id

```
r : receive (at to_node)
+ : enqueue (at queue)              src_addr : node.port (3.0)
- : dequeue (at queue)              dst_addr : node.port (0.0)
d : drop    (at queue)
r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
+ 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
- 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
- 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

3.2 Delay

Delay is simply the time that elapsed during the travel of a packet between two points. This travel could either be from a source premise to destination premise or from network ingress to network degrees. The more the value for the delay grows larger, so it poses more difficulty for the transport layer protocols to sustain high bandwidths. These delay characteristics which includes average delay, delay bound and variance of delay/jitter, etc has number of several ways they can be specified .In this research, end to end delay was calculated.

3.3 Loss of Packet

When a queue linked to a node in the network overflows, it may lead to lost in network Packets. The measure of loss of packet in the midst of the reliable state is another basic property in the scheme of congestion control. It poses more difficulty for high bandwidths to be maintained by transport-layer protocols if the packet loss value grows larger. The individual applications strongly determine the sensitivity it losses packets, and the patterns and also the frequency of its loss among longer packet sequences. There are different other ways to specify this characteristic which includes: rate of loss, loss free seconds, loss patterns, and conditional loss probability. In this research, only loss due to packet dropping of the packets was considered.

3.4 Queue Length

The queue length measures how efficient and effective a congestion control algorithm's active queue management has been performing and so is an important characteristic. Queuing system in networks can be seen as either arriving of packets for service; if its not immediate, waiting for service, and if packets have waited for service, leaving the system after being served.

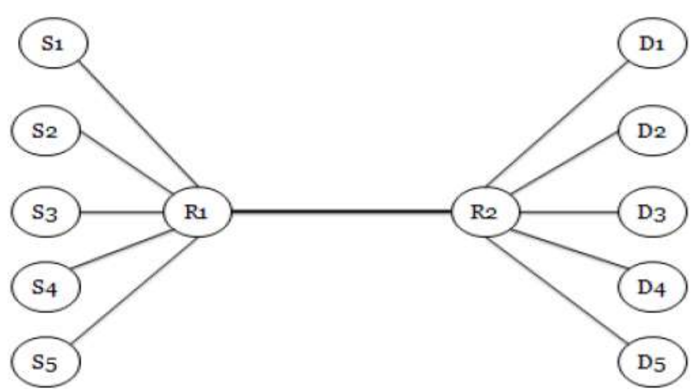
3.5 Throughput

Throughput is the most widely used, and is referred to as the characteristic of the basic performance measure. It describes the measure of how quick the receiver is capable of getting a certain amount of data the sender sent. It is an important factor that impacts directly on the performance of the network. It is determined as, the ratio of the end to end delay to the total amount of data received.

4.0 Simulations and its Results

In this section, network configuration were discussed ;being used over a network simulator known as **Ns2,** to do the simulation of the above mentioned three algorithms under consideration which are: RED, REM and SFQ. And after which the results we obtained from our simulations were analyzed. The algorithms compared here were firstly deployed into the simulator architecture, and then the following simulation scenario was generated with an aim to compare their different performance against a simulation setting as Figure3 displays.

Fig. 3 Scenario for Simulation



4.1 Simulation Scenario

At the bottleneck link, each side has five nodes. The five nodes here are all behaving as a TCP source and the five nodes are responding as a TCP link so that both routers are applying the congestion control algorithm. There is two- way traffic in the system. Considering the network scenario as shown in Figure 3. This network was simulated on ns2 for different AQM algorithms

(RED, SFQ and REM) for same network parameters as given in Table 1 except to the bottleneck link. We simulated on the same bottleneck link R_1R_2 these three algorithms. Firstly, 5Mbps was considered the bottleneck link for each considered AQM algorithm. We also considered a fixed packet size of 2 KB and a buffer capacity of 4KB throughout the simulation. In Table 1, the Round trip delay for each of the link has been displayed. Therefore it could be concluded from the Table that a minimum end to end delay should be larger than 60ms. Our simulation has been observed over the period of 100 seconds.

4.2 Analysis of the Loss Rate

In our simulation, the bandwidth of the bottleneck link was varied as given in Figure 4 for each algorithm. This Figure 4 shows the loss rate as occurred in RED, SFQ, and REM respectively. It has been shown that the rate of loss smoothly decreased as we are increasing the bandwidth of bottleneck link in the case of RED. A drastic change in loss rate at 15 Mbps was got in the case of SFQ and this is because of unfairness realized at this said bandwidth. It has been concluded that at higher bandwidth SFQ and REM could achieved higher loss rate. But the RED over an increase in bandwidth shows smooth decrease in loss rate.

4.3 The Analysis for Queue length

In terms of the analysis for queue length, there was no significant difference between the considered algorithms; this is due to the fact that only two packets at most could be allowed to enter into queue because of the little capacity of the buffer. As shown in Figure 7, **REM** achieved for a longer time queue length of just two packets.

4.4 The Throughput Analysis

It has been well observed that the REM had the best ***throughput*** and that the RED had the least throughput among the three considered algorithms for the simulation achieved at 5 Mbps of bandwidth. Figure 5 also shows that REM gets the good result and RED gets the poor result. It could be observed that at one point on the throughput graph that whenever smooth growth in throughput has been broken, it indicated a starting point when dropping of packet took place. This point achieved in each of the algorithm has same ratio as compared to their maximum achieved throughput.

4.5 The Delay Analysis

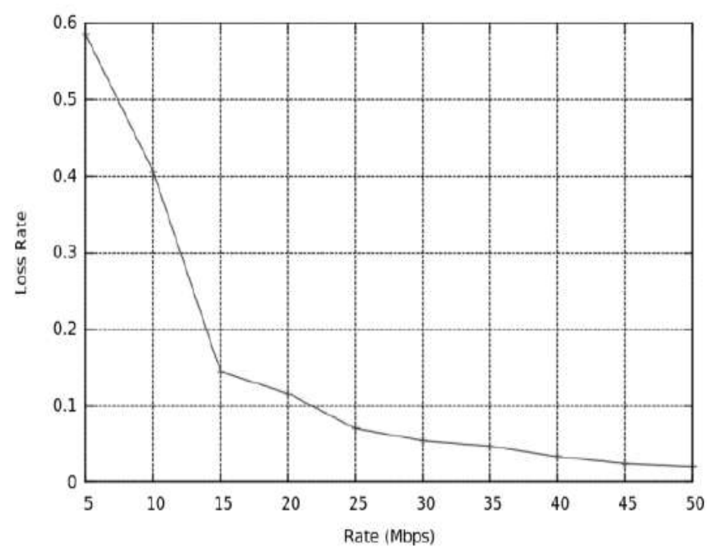
In figure 6, the exact response time that each packet achieved in each of the considered algorithms was plotted. Analysis from Table 2 shows that maximum delay was achieved in the REM; although each of the algorithms recorded the occurrence of same minimum delay. Therefore, it could be reasoned conclusively that each of the considered algorithm will arrive at same response time as long as congestion has been detected; This is because queuing delay would be same for each algorithm if no congestion in the network.

Table 1:. Simulation Parameters

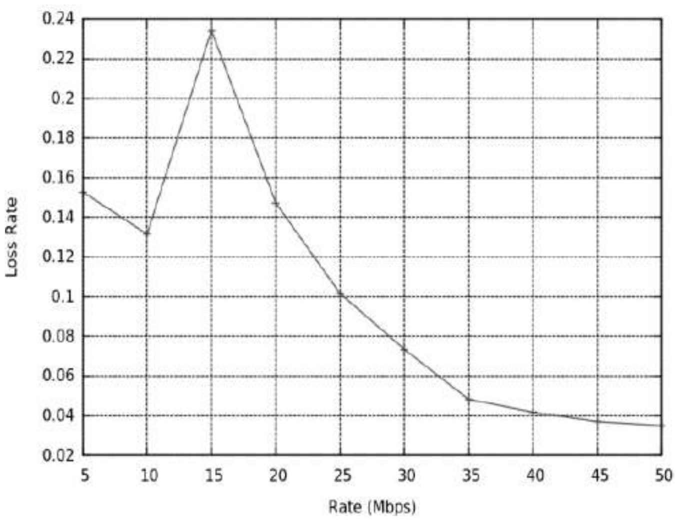
Link	*RTT (ms)*	*Rate (Mbps)*	*Protocol*
S1 R1	10	100	Drop tail
S2 R1	10	100	Drop tail
S3 R1	10	100	Drop tail
S4 R1	10	100	Drop tail
S5 R1	10	100	Drop tail
R1R2	40	10	RED / SFQ /REM
R2D1	10	100	Drop tail
R2D2	10	100	Drop tail
R2D3	10	100	Drop tail
R2D4	10	100	Drop tail
R2D5	10	100	Drop tail

Fig. 4. Loss Rate for the three considered algorithms

(a) RED



(b) SFQ



(c) REM

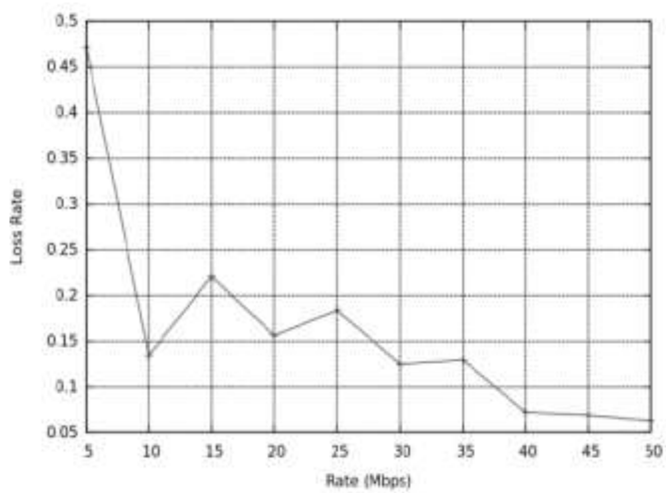
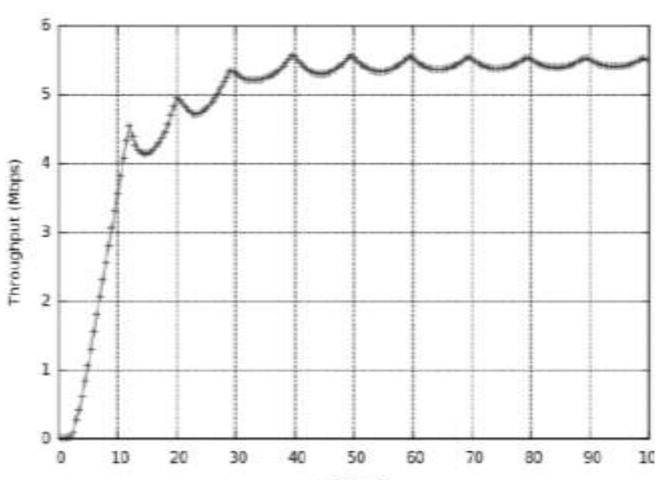
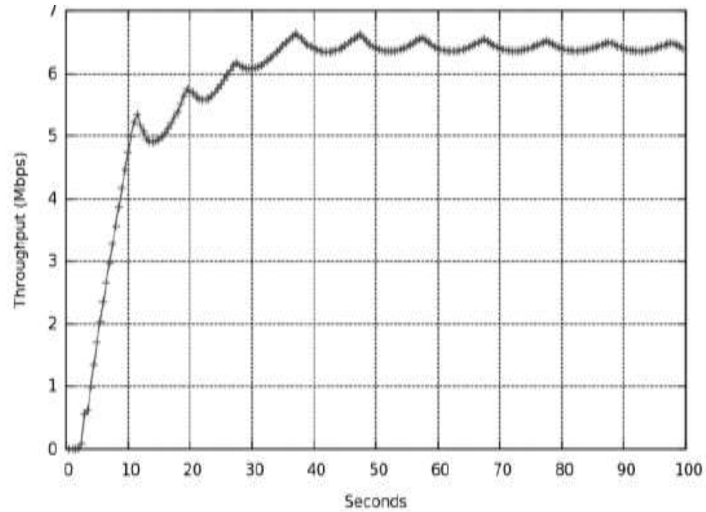


Fig. 5. Throughput Diagram for the three considered algorithms

(a) RED



(b) SFQ



(c) REM

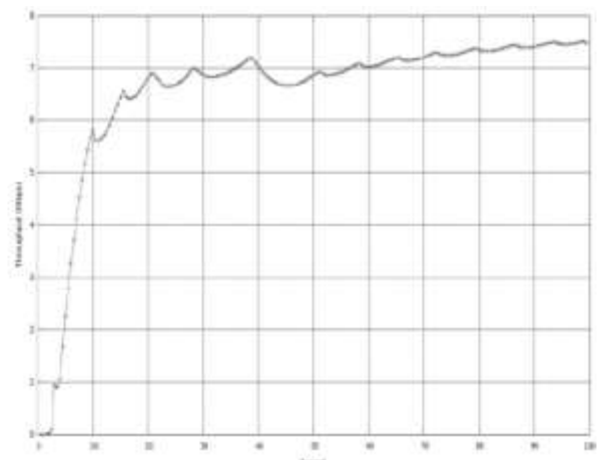
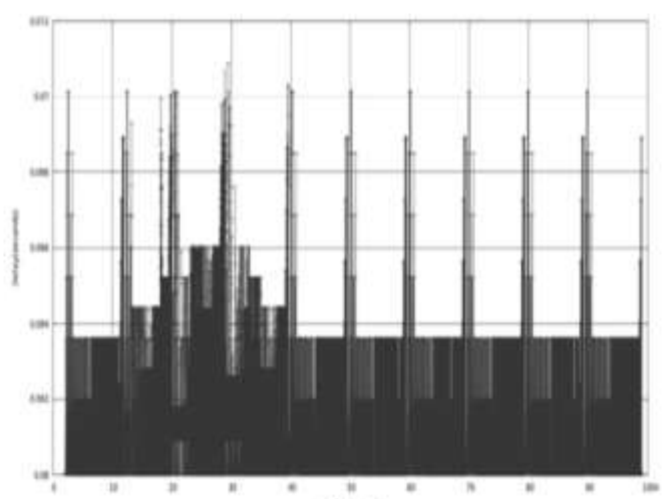
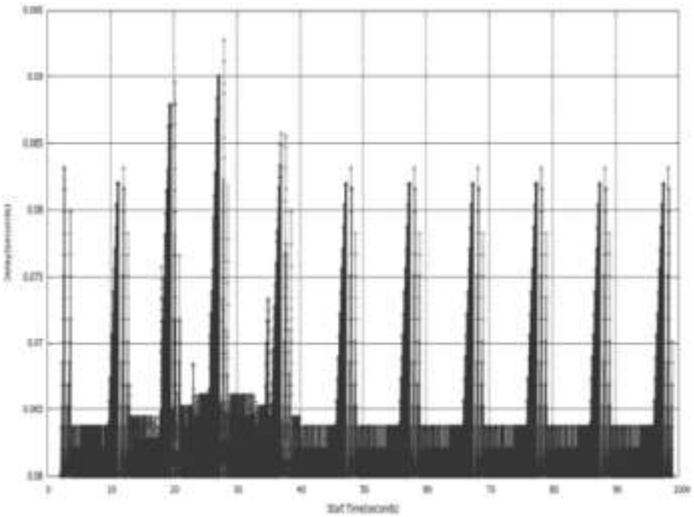


Fig. 6. The Delay Diagram for the three considered algorithms

(a) RED



(b) SFQ



(c) REM

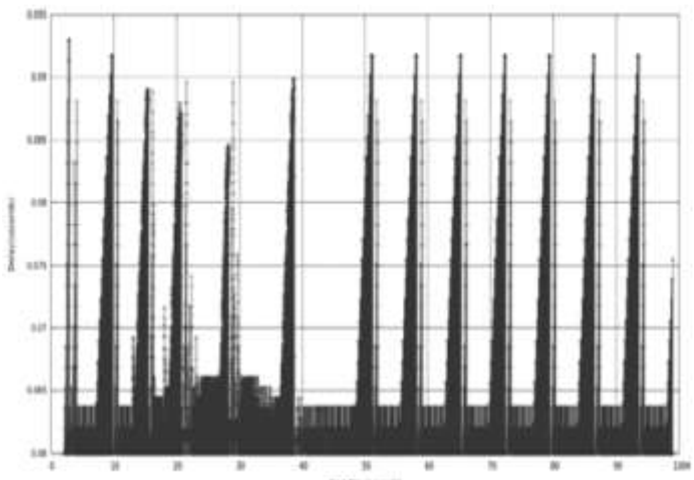
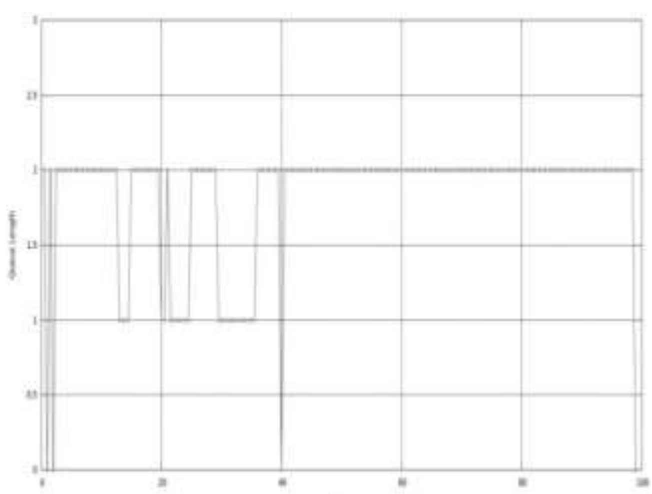
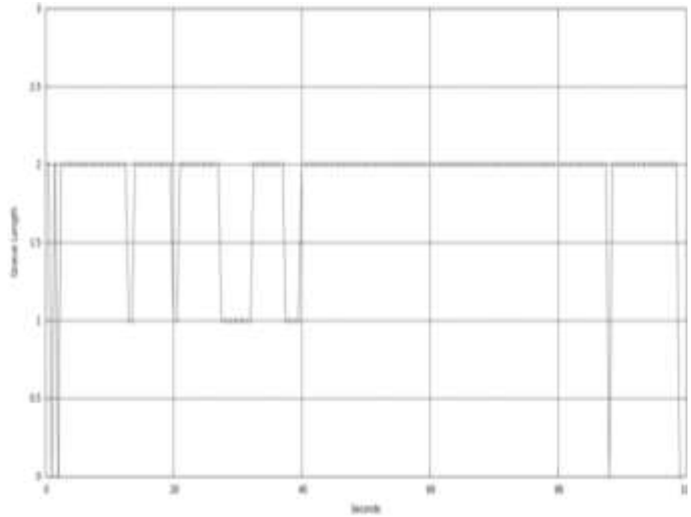


Fig. 7. Queue Length Diagram for the three considered Algorithms

(a)RED



(b) SFQ



(c) REM

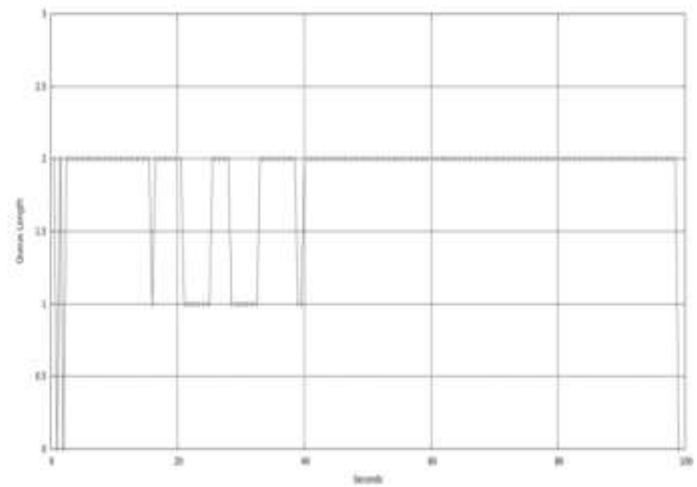


Table 2. The Comparative results

Performance Metrics		RED	SFQ	REM
Queue length	Max.	2	2	2
	Min.	0	0	0
Throughput	Max.	5.53	6.64	7.51
	Min.	0	0	0
Delay	Max.	67.25	90.01	92.96
	Min.	60.03	60.03	60.03
Send Packets		37157	42554	49117
Lost Packets		151	56	66
Average Loss Ratio (%)		0.4064	0.1316	0.1344
Utilization (%)		59.45	68.08	78.58

Table 3. The different algorithms rankings

Algorithm	Delay	Queue Length	Throughput	Loss Rate
RED	A	A	C	C
SFQ	B	B	B	A
REM	C	C	A	B

5.0 Conclusion

In this research work, the problems associated with existing congestion control algorithms was addressed and as we tried showing the performance parameters of **RED**, **REM**, and **SFQ** for the network configurations considered. The various parameters of performance for each considered algorithm was calculated as Table 1and Figure 3 shows. The total amount of packets deployed over bottleneck link R_1R_2 was also calculated and the total amount of packets lost in a period of 100 seconds during the simulation. **RED** posses a maximum ratio of loss while **SFQ** posses minimum average ratio of loss. Now the real amount of data bytes sent over the R_1R_2 bottleneck link could be calculated (known as utilization) as Table2 shows. Observations also show that there is variation according to each algorithm in terms of performance parameters. In terms of loss ratio, throughput and utilization, **REM** achieved best of result. Although **RED** in terms of delay was better off. If provided with equal weightage to each performance parameter , then we could conclude that among considered three algorithms under our simulation, that **REM** is the best one. As Table 3 displays rankings for their parameter of performance, A indicates a higher ranking and a decrease up to C.

We plan in future work to develop hybrids of each of these algorithms to achieve better results and also be able to extend the simulation for the hybrids that will comprise their various advantages.

REFERENCES

[1]. Chughtai, O., Badruddin, N., Rehan, M., Khan, A., (2017) Wireless Communication and Mobile Computing, Congestion detection and alleviation in multihop wireless sensor networks, vol. () pp. 13. https://doi.org/10.1155/2017/9243019

[2]. D.Lin and R. Morris, (2017) “Dynamics of random early detection,”, ACM SIGCOMM Computer Communication Review, vol. 27,no.4, pp.127-137.

[3]. Essays, UK (2013). Studying the Causes of Network Congestion Information Technology Essay, Retrieved from https://www.ukessays.com/essays/information-technology/studying-the-causes-of-network-congestion-information-technology-essay.php?vref=1,datapath.io/resources/blog/10-**causes-of-network**

[4]. IBM knowledge center, (2012). “Congestion Management. Designing a low latency messaging”.

[5]. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1, September 2011 ISSN (Online): 1694-0814 www.IJCSI.org 94

[6]. Logan, R., (2016). The negative effect of network congestion problem

[7]. Network Simulator Manual, http:www.isi.edu/nsam/ns/index.html.

[8]. P.E. Meckenney, (1990) “Stochastic Fair Queuing”, In proceedings of INFCOM, vol. 2, pp. 733-740,.

[9]. Prateek Mehrotrat, et al, **(**2011) **“**Comparative Analysis of Congestion Control Algorithms Using ns-2” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1, ISSN (Online): 1694-0814 www.IJCSI.org

[10]. Prabakaran, N., Prabakaran, R., Shanmuga, R. B., Sarma Dhulipala, V.R., (2011). International Conference on Devices and Communications, Rate optimization scheme for node- level congestion in wireless sensor networks. DOI: 10.1109/ICDECOM.2011.5738548

[11]. S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. IEEE/ACM Trans. On Netwoking, vol.1, no.4, pp.397–413, August 1993.

[12]. W. Feng, K. G. Shin,, D. D. Kandlur, , and D. Saha,, (2002); “The BLUE Active Queue Management Algorithms,” IEEE/ACM Transaction on Networking, vol. 10, no. 4, pp.513 – 528.

[13]. S.Athuraliya, Victor H. Li, Steven H. Low and Qinghe YinK. Elissa, “REM: Active Queue Management,” *IEEE Network*, vol. 15 ,no.3, pp. 48 – 53, August 2002.

[14]. S. Patel, P. Gupta and G. Singh,(2010) “ Performance Measure of Drop tail and RED,” In proceedings of *ICECT* , pp. 35 –38.

[15]. Palanisamy, A., (2016). “When does network congestion occur?” www.bayt.com/.../when-does-network-congestion-occur

[16]. Thrimoorthy, N., Dr. Anuradha, T., (2016) International Journal of Engineering Research and Development, congestion detection approaches in wireless sensor networks: A comparative study, Vol. 12 (3) 59-63