*International Refereed Journal of Engineering and Science (IRJES)*
*ISSN (Online) 2319-183X, (Print)* 2319-1821
*Volume 1, Issue 2 (October 2012), PP.41-45*
*www.irjes.com*

# Congestion Control in Communication Network Using RED, SFQ and REM Algorithm

Dolly Kalav[1], Sudha Gupta[2]
[1]*(Department of Electronics Engineering, Mumbai University, India)*
[2]*(Department of Electronics Engineering, Mumbai University, India)*

**ABSTRACT :** *This paper is an exploratory survey of TCP congestion control principles and techniques. By studying congestion control techniques used in TCP implementation software and network hardware we can better comprehend the performance issues of packet switched networks and in particular, the public Internet. Interaction between Transmission Control Protocol (TCP) and Random Early Detection (RED) gateways can be captured using dynamical models. In order to curtail the escalating packet loss rates caused by an exponential increase in network traffic, active queue management techniques such as Random Early Detection (RED) have come into picture. Stochastic Fair Queuing (SFQ) ensures fair access to network resources and prevents a busty flow from consuming more than its fair share. In case of (Random Exponential Marking) REM, the key idea is to decouple congestion measure from performance measure (loss, queue length or delay). Performance parameter of RED, SFQ and REM algorithm is analysis using NS-2 network simulator.*

***Keywords***- *Active Queue Management (AQM), Congestion Control, Hybrid System, Open Systems Interconnection (OSI), Random Early Detection (RED), Random Exponential Marking (REM), Stochastic Fair Queuing (SFQ), Stochastic Processes, Transmission Control Protocol /Internet Protocol (TCP/IP).*

## I. INTRODUCTION

A network is combination of hardware and software that sends data from one location to another. The hardware consists of the physical equipment that carries signals from one point of the network to another. The software consists of instruction sets that make possible the services that we expect from a network. The Open Systems Interconnection (OSI) model is a reference tool for understanding data communications between any two networked systems. TCP (Transmission Control Protocol) is responsible for verifying the correct delivery of data from client to server. Data can be lost in the intermediate network. TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received. IP (Internet Protocol) is responsible for moving packet of data from node to node. IP forwards each packet based on a destination IP address.

SFQ (Stochastic Fair Queuing) is a class of queue scheduling disciplines that are designed to allocate a pretty large number of separate FIFO queues. Increasing the number of queues to a large extent helps to achieve fairness. RED queue management aims at alleviating this problem by detecting incipient congestion in advance and communicating the same to the end-hosts, allowing them to trim down their transmission rates before queues begin to overflow and packets start dropping. For this, RED maintains an exponentially weighted moving average of the queue length which it used as a congestion detection mechanism. In order to be efficient, RED must ensure that congestion notification is conveyed at a rate which sufficiently suppresses the transmitting sources without underutilizing the link. RED must also ensure that the queue is configured with enough buffer space to hold an applied load greater than the link capacity from the time when congestion detection occurs to the time when the applied load reduces at the bottleneck link in response to the notification regarding congestion. When a flow persistently occupies a considerable amount of the queue’s buffer space, it is identified and restrained to a smaller buffer space. Severity of congestion is indicated by queue lengths in various queue management algorithms. This inherent problem can be dealt by a fundamentally different active queue management algorithm, called BLUE. BLUE has been shown to perform significantly better than RED both in terms of packet loss rates and buffer size requirements in the network. If buffer overflow causes the queue to recurrently drop packets, BLUE increments the marking probability, thus augmenting the rate at which congestion notification is sent back.

REM is an active queue management scheme that aims to achieve both high utilization and negligible loss and delay in a simple and scalable manner. The first idea of REM attempts to match user rates to network capacity while clearing buffers, irrespective of number of users. The second idea embeds the sum of link prices (congestion measures), summed over all the routers in the path of the user to the end-to-end marking (or dropping) probability. Number of active flows shares a linear relationship with number of different flows in the buffer. We simulated the network configuration having higher delay and lower bandwidth at the main bottleneck link. In this paper, we are using ns-2 network simulator.

## II. ANALYSIS TRACE FILE

When the ns are run, the trace of each event can be stored in a trace file. While tracing into an output ASCII file, the trace is organized in 12 fields as shown in the following figure 1.

| event | time | from node | to node | pkt type | pkt size | flags | fid | src addr | dst addr | seq num | pkt 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)
```

+ 160.012933 1 0 tcp 60 ------- 2 1.0.1.0 0.0.0.0 0 64
- 160.012933 1 0 tcp 60 ------- 2 1.0.1.0 0.0.0.0 0 64
r 160.015029 1 0 tcp 60 ------- 2 1.0.1.0 0.0.0.0 0 64
+ 160.015029 0 1 ack 40 ------- 2 0.0.0.0 1.0.1.0 0 65
- 160.015029 0 1 ack 40 ------- 2 0.0.0.0 1.0.1.0 0 65

**Fig. 1 Trace files structure**

### 2.1 Packet Loss

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss is distinguished as one of the three main error types encountered in digital communications; the other two being bit error and spurious packets caused due to noise.

Packets can be lost in a network because they may be dropped when a queue in the network node overflows. The amount of packet loss during the steady state is another important property of a congestion control scheme. The larger the value of packet loss, the more difficult it is for transport-layer protocols to maintain high bandwidths, the sensitivity to loss of individual packets, as well as to frequency and patterns of loss among longer packet sequences is strongly dependent on the application itself.

### 2.2 Throughput

This is the main performance measure characteristic, and most widely used. In communication networks, such as Ethernet or packet radio, throughput or network throughput is the average rate of successful message delivery over a communication channel. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot.

This measure how soon the receiver is able to get a certain amount of data send by the sender. It is determined as the ratio of the total data received to the end to end delay. Throughput is an important factor which directly impacts the network performance.

### 2.3 Delay

Delay is the time elapsed while a packet travels from one point e.g., source premise or network ingress to destination premise or network degrees. The larger the value of delay, the more difficult it is for transport layer protocols to maintain high bandwidths. We will calculate end to end delay.

### 2.4 Queue Length

A queuing system in networks can be described as packets arriving for service, waiting for service if it is not immediate, and if having waited for service, leaving the system after being served. Thus queue length is very important characteristic to determine that how well the active queue management of the congestion control algorithm has been working.

## III. SIMULATIONS AND RESULTS

In this section, we discussed about network configuration used over the network simulator ns2 to simulate the three algorithms RED, SFQ and REM and after that we analyzed about the results obtained from our simulations.

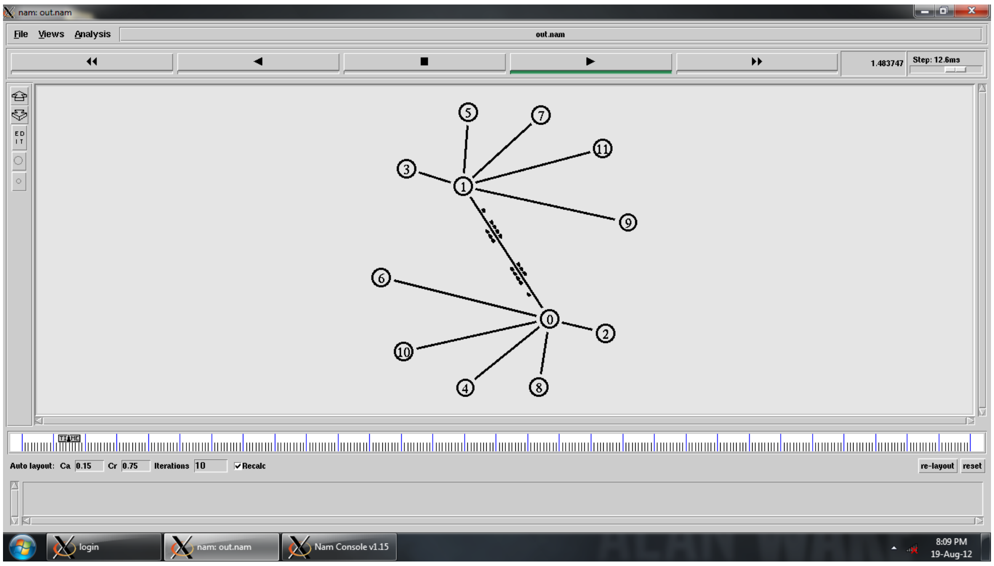


**Fig. 2 Simulation Scenario**

### 3.1 Simulation Scenario

There are eight nodes at each side of the bottleneck link. Here eight nodes are acting as a TCP source and eight nodes are acting as a TCP sink so that both routers are applying the congestion control algorithm. There is two- way traffic in the system. We consider the network scenario as shown in Figure 2. We simulate this network 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 these three algorithms RED, SFQ, and REM on the same bottleneck link node 8 and node 9. Firstly we consider the bottleneck link to 5Mbps for each considered AQM algorithm. We considered a fixed packet size of 5 KB and buffer capacity of 8KB throughout the simulation. Round trip delay for each link has been displayed in Table 1. So it could be concluded from the Table 1 that minimum end to end delay should be larger than 160 ms. this simulation has been observed over the period of 100 seconds. Whole simulation has been observed over small buffer capacity of 4KB.

**Table 1:.Parameters for simulation**

| *Link* | *RTT (ms)* | *Rate (Mbps)* | *Protocol* |
|---|---|---|---|
| S1 R1 | 40 | 5 | Drop tail |
| S2 R1 | 40 | 5 | Drop tail |
| S3 R1 | 40 | 5 | Drop tail |
| S4 R1 | 40 | 5 | Drop tail |
| S5 R1 | 40 | 5 | Drop tail |
| S6 R1 | 40 | 5 | Drop tail |
| S7 R1 | 40 | 5 | Drop tail |
| S8 R1 | 40 | 5 | Drop tail |
| R1R2 | 80 | 10 | RED/SFQ/REM |
| R2D1 | 40 | 5 | Drop tail |
| R2D2 | 40 | 5 | Drop tail |
| R2D3 | 40 | 5 | Drop tail |
| R2D4 | 40 | 5 | Drop tail |
| R2D5 | 40 | 5 | Drop tail |
| R2D6 | 40 | 5 | Drop tail |
| R2D7 | 40 | 5 | Drop tail |
| R2D8 | 40 | 5 | Drop tail |

**3.2 Analysis of Loss Rate**

Figure 3 shows about the loss rate occurred in RED, SFQ, and REM respectively. In our simulation, we vary the bandwidth of the bottleneck link as given in Figure 4 for each algorithm RED, SFQ, and REM. It has been observed that loss rate smoothly decreased as we are increasing the bandwidth of bottleneck link in case of RED. We got the drastic change in loss rate at 15 Mbps in case of SFQ because of unfairness achieved at this bandwidth. It has been concluded that SFQ and REM could achieved higher loss rate at higher bandwidth at some specific bandwidth but it could not be happen. It has been reflected more in case of SFQ. But RED shows smooth decrease in loss rate over increase in bandwidth.

**3.3Analysis of Throughput**

It has been observed from Fig. 4 that REM had a best throughput and RED had least throughput among all these three algorithms for the simulation achieved at 5 Mbps of bandwidth. Figure 5 show that REM gets the good result and RED gets the poor result. It could be observed one point on throughput graph whenever smooth growth in throughput has been broken. It indicated about a starting point when dropping of packet took place. This achieved point in each algorithm has a same ratio as compared to their maximum achieved throughput.

**3.4Analysis of Delay**

Figure 5 plots the actual response time for each packet achieved in RED, SFQ, and REM. It has been observed from Table 2 that minimum delay occurred in each algorithm is same but maximum delay achieved in REM. Therefore we could conclude that each algorithm would get a same response time provided congestion has been observed because queuing delay would be same for each algorithm if there is no congestion in network.

**3.5 Analysis of Queue length**

Here we did not achieve much difference in queue length between these algorithms because at most two packets could be allowed to enter into queue due to the small buffer capacity. REM achieved queue length of two packets for a longer time as shown in Figure 6.

## IV. CONCLUSION AND FUTURE WORK

Research effort has been focused on understanding and utilizing RED algorithm to leverage the current network. For example, since it is widely accepted that Poisson model is not sufficient to characterize the traffic in current Internet, it is important to understand how RED and similar Active Queue Management (AQM) algorithm act when self-similar network traffic is applied.

In this paper we address the problems with existing congestion control algorithms and we tried to show about various performance parameters of RED, SFQ, and REM for our considered network configurations. We have calculated the different performance parameters for each algorithm of considered network configuration as given in Figure 2 and Table 1.

**Table 2 Comparative results**

| Performance | RED | SFQ | REM |
|---|---|---|---|
| Queue length (Max) | 4 | 4 | 4 |
| Throughput (Max) | 6.63 | 7.89 | 9.10 |
| Delay (Max) | 75.75 | 110.01 | 112.25 |
| Send Packets | 38515 | 45524 | 50114 |
| Lost Packets | 162 | 62 | 92 |
| Average Loss Ratio (%) | 0.5267 | 0.2143 | 0.2156 |
| Utilization (%) | 60.21 | 79.54 | 89.41 |

**Table 3 Ranking of the different algorithms**

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

We calculated the total number of packets sent over the bottleneck link node 8 & 9 and total number of packets lost during the simulation over the period of 100 seconds. SFQ has a minimum average loss ratio and RED has a maximum loss ratio. Now actual number of bytes transmitted over the bottleneck link node 8 & 9 could be computed termed as utilization has been shown in Table 2. It has been observed that performance parameters are varying according to the algorithms. RED achieved the best result in terms of the delay but in terms of throughput, loss ratio, and utilization REM shows the best results. If we would provide the equal weight age to each performance parameter then we could conclude that REM would is the better one among all three algorithms considered in our simulation.

## V. Acknowledgements

I am profoundly grateful to my project guide, **Mrs. Sudha Gupta**, for giving valuable advice and enthusiastic guidance. I express my gratitude to all the faculties for the interest she has shown in this project.

I am also thankful to **Prof. Pranav Pawar**, who devoted his valuable time and helped me in all possible ways towards successful completion of this work.

Last but not the least, I wish to avail myself of this opportunity, to express a sense of gratitude and love to my friends and my family members for their manual support, strength, help and for everything.

## REFERENCES

[1] Stefan kohler, Michael Menth and Norbert Vicari, “Analytic Performance Evaluation of the RED Algorithm for Qos in TCP/IP Networks”, 9th IFIP Conference on Performance Modeling and Evaluation of ATM & IP Networks 2001, Budapest.
[2] Wu-chang Feng, Mamber, ACM, Kang G. Sin, Fellow, IEEE and ACM, Dilip D. Kandlur, Member, IEEE, Debanjan Saha, Member, IEEE, “The BLUE Active Queue Management algorithms”.
[3] Amit Aggarwal, Stefan Savage, and Thomas Anderson, “Understanding the Performance of TCP Pacing”, IEEE Info Com 2000 March 30, 2000.
[4] Raj Jain, K. K. Ramakrishanan, “Congestion Avoidance in Computer Networks with a Connectionless Networks Layer”, Digital Equipment Corporation, September, 1998
[5] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, d. Estrin, S. Floyd, v. Jacobson, G. Minshall, C. Patridge, L. Peterson, K. Ramakrishan, S. Shenker, J. WrocLawski, and Lixia Zhang, “ Recommendations on Queue Management and Congestion Avoidance in the Internet”, April 1998, RFC 2309.
[6] S. Athuraliya, S. Low, V. Li, and Q. Yin, “REM: Active Queue Management”, IEEE Network Magazine, May 2001.
[7] W. Fung, D. Kandlur, D. Saha, and K. G. Shin, “Stochastic Fair Blue: A Queue Management Algorithms for Enforcing Fairness”, In Proc. IEEE INFOCOM, pages 1520-1529, April 2001.
[8] S. Floyd and v. Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, ACM/IEEE Transactions on Networking, 397-413, August 1993.
[9] P. McKenney, “Stochastic Fairness Queuing”, In Proc. IEEE INFOCOM, March 1990.
[10] T. Bhaskar Reddy and Ali Ahammed, “Performance Comparison of Active Queue Management Techniques”, Journal of Computer Science, Pages 1020-1023, 2008.
[11] Van Jacobson, “Congestion Avoidance and Control”, in SIGCOMM’88, pages 314-329, August 1988.
[12] V. Firoiu, M. Borden, “A study of Active Queue Management for Congestion Control”, IEEE INFOCOM, 2000.
[13] Hui Zhang, Mingjian Liu, Vladimir Vukadinovic, and Ljiljana Trajkovic, “Modeling TCP/RED: A Dynamical Approach”.
[14] Andrew S. Tanenbaum, “Computer Networks”.
[15] Behrouz A Forouzan, “Data Communications and Networking”.