A Comparative Analysis of Queue Management Techniques using NS-2 Simulator

P. T. Mahida
Department of IT,
SVMIT, Bharuch

Kinjal Patel
Department of IT,
SVMIT, Bharuch

Nayan Vanza
Department of IT,
SVMIT, Bharuch

Siddharth Patel
Department of It,
SVMIT, Bharuch

ABSTRACT

Effectively and fairly allocating resources to the competing users in a network is a major issue to meet the demand for higher performance nowadays. Queue management enhances the efficiency of transfers and cooperates with Transmission Control Protocol (TCP) in adapting the intense flow of the congestion in the network. The shared resources of a network are bandwidth of the link and queues on the routers and switches. As too many packets are queued awaiting transmission, the queues overflow and the packets have to be dropped which results into congestion. The queue management algorithm, which is applied to a router, plays an important role in providing Quality of Service (QoS). In this paper, we have presented a simulation based comparison and evaluation of four popular queue management schemes: Stochastic Fair Queuing (SFQ), Random Early Detection (RED), Random Exponential Marking (REM) and Droptail in terms of packet drop rate and delay. Simulation is done using Network Simulator (ns2.34) Our Simulation results indicate that REM performed better in terms of packet drop rate and RED performs better in terms of end-to-end delay.

Keywords

RED, SFQ,REM, Droptail,, Packet drop rate, ns2.

1. INTRODUCTION

The Internet traffic generates stream of data packets in the network with different traffic profiles and leads to congestion. Congestion refers to a network state where a node or link carries so much data that it may decrease network service quality, resulting in queuing delay, frame or data packet loss and blocking of new connections [1]. Congestion is an important issue which researchers focus on in the *TCP* network environment. To control the congestion, there are two types of algorithms: Source-side Algorithm and Sink side Algorithm. Source side Algorithm is sender side and another is sink side algorithm. To keep the stability of the whole network, congestion control algorithms have been extensively studied. Queue management method employed by the routers is one of the important issues in the congestion control study. During congestion, large number of packets, face delay or even get dropped due to queue overflow. As a result congestion results in degradation of the throughput and large packet loss too. Due to very high traffic load, the performance degrades completely and almost no packets are delivered to sink nodes. To resolve the problem, many congestion control algorithms [2,3] are proposed. Many of the algorithms are based on the evaluation of the feedback from the congested network. Some algorithms detect congestion from warn packets sent back to the source while in other sources observe change in few network parameter like delay, packet drop and detect congestion[4]. In queue management algorithms there are three different types of algorithms, active, passive and proactive like DropTail, SFQ, RED and REM. Drop Tail is the most widely used queue management method in today's IP networks. RED is mostly the default method implemented in the routers nowadays. RED monitors the average queue size and drops packets based on statistical probabilities.

SFQ is a simple implementation of the fair queuing algorithms family. It's less accurate than others, but it also requires less calculation while being almost perfectly fair. 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 [5, 6]. We have analyzed performance of different queue management algorithms by applying them on different simulation scenario at different transfer rate of packets.

In this paper, we will compare popular Queuing Management Techniques, Random Early Detection [7], DropTail, Random Exponential Marking (REM)[3] and Stochastic Fair Queuing (SFQ) in different aspects , such as delay and Packet Drop Rate.

In section II, we have given overview of Queue management techniques. Section III describes OPNET implementation and simulation model and topology. Section IV gives performance comparisons with various queue management techniques by simulation in ns2. Conclusion is presented in section VI.

2. QUEUE MANAGEMENT TECHNIQUES

Queue management is defined as the algorithm that manages the length of packet queues by dropping packets when necessary or required to be dropped. From the point of dropping packets, queue management can be classified into three categories as in the figure [7].

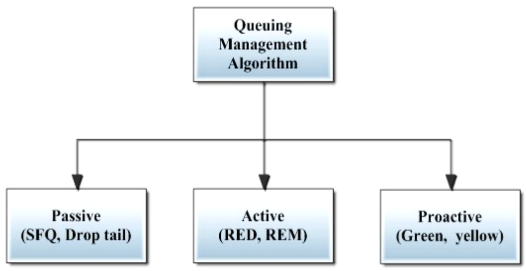


Fig 1: Queue Managment Techniques

Active queue management is expected to eliminate global synchronization and improves quality of service. The expected advantages of active queue management increases the throughput, reduces delay, and avoides lock-out. Random Early Detection (RED), an active queue management scheme, has been recommended by the Internet Engineering Task Force (IETF) as a default active queue management scheme for next generation networks

2.1 Passive Queue Management

In Passive Queue management (PQM) technique, an Internet router typically maintains a set of queues, one per interface, that hold packets scheduled to go out on that interface. Such queues use a *drop-tail* discipline: a packet is put onto the queue if the queue is shorter than its maximum size (measured in packets or in bytes), and dropped otherwise.PQM does not employ preventive packet drop before the router buffer gets full.

Droptail: In Droptail, the router accepts and forwards all the packets that arrive as long as its buffer space is available for the incoming packets. If a packet arrives and the queue is full, the incoming packet will be dropped. The sender eventually detects the packet lost and shrinks its sending window. Drop-tail queues have a tendency to penalize bursty flows, and to cause global synchronization between flows [8]

2.2 Active Queue Management

In Internet routers, active queue management (AQM) is a technique that consists in dropping or ECN-marking packets before a router's queue is full. Typically, they operate by maintaining one or more drop/mark probabilities, and probabilistically dropping or marking packets even when the queue is short[8].

RED: RED is a type of active queue management technique used for congestion avoidance. RED monitors the average queue size and drops (or marks when used in conjunction with ECN) packets based on statistical probabilities. If the buffer is almost empty, all

incoming packets are accepted. As the queue grows, the probability for dropping an incoming packet grows too. When the buffer is full, the probability has reached 1 and all incoming packets are dropped. RED is more fair than tail drop, in the sense that it does not possess a bias against bursty traffic that uses only a small portion of the bandwidth. The more a host transmits, the more likely it is that its packets are dropped. The probability of a host's packet being dropped is proportional to the amount of data it has in a queue. Early detection helps avoid TCP global synchronization

REM: REM is an active queue management scheme that measures congestion not by performance measure such as loss or delay, but by quantity. REM can achieve high utilization, small queue length, and low buffer overflow probability. Many works have used control theory to provide the stable condition of REM without considering the feedback delay. In case of (Random Exponential Marking) REM, the key idea is to decouple congestion measure from performance measure (loss, queue length or delay). In REM, the user rates are matched by clearing buffers irrespective of number of users. The sum of link prices, summed over all the routers in the path of the user to the end-to-end marking [9]

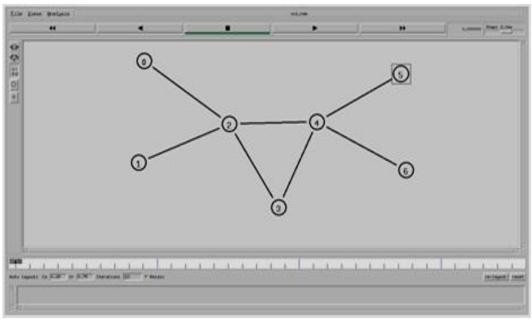
2.3 Pro-active Queue Management

Pro-active queue management algorithms are novel attempts to prevent congestion from ever happening in the first place. e present a proactive queue-management (PQM) algorithm called GREEN that applies knowledge of the steady state behavior of TCP connections to intelligently and proactively drop packets, thus preventing congestion from ever occurring and ensuring a higher degree of fairness between flows. This congestion-prevention approach is in contrast to the congestion avoidance approach of traditional active queue-management schemes where congestion is actively detected early and then reacted to. In addition to enhancing fairness, GREEN keeps packet-queue lengths relatively low and reduces bandwidth and latency jitter. These characteristics are particularly beneficial to real-time multimedia applications. Further, GREEN achieves the above while maintaining high link utilization and low packet loss [10]

3. SIMULATION MODEL AND ENVIRONMENT

The experiments were conducting using the ns-2 network simulator. There are seven nodes in the experiment conducted. Few nodes are acting as a TCP sink node and few as source node. There is a 2-way traffic in the system..We have simulated the scenario on network on ns2 for different algorithms like SFQ, RED and REM for bottleneck link and Droptail for other link.

We have simulated these algorithms using the figure 2 and 3. We have simulated the scenario by varying the data rate from 3 to 30 Mbps. As the data rate increases, we have observed an improvement in the results.



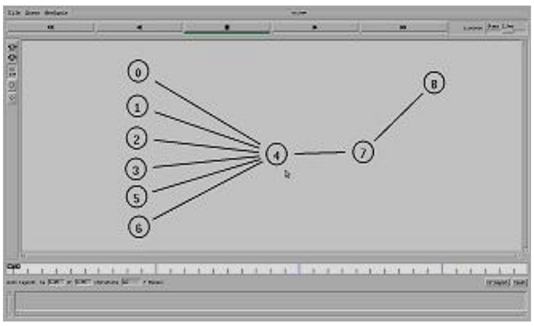


Figure 3: Simulation Topology 2

3.1 Simulation Metrics

We have simulated four algorithms of the Passive and Active queue management algorithms like Droptail, SFQ, RED and REM. Firstly, we have varied the data rate and observed the results. Then, the simulation is done by varying the data rate and observed the Packet loss rate and End-to-end delay. The simulation metrics are explained below.

3.2 Simulation Metrics

We have simulated four algorithms of the Passive and Active queue management algorithms like Droptail, SFQ, RED and REM. Firstly, we have varied the data rate and observed the results. Then, the simulation is done by varying the data rate and observed the Packet loss rate and End-to-end delay. The simulation metrics are explained below:

3.2.1 Data Rate

The data rate is the amount of data that is moved from one place to another in a given time. In network, the data rate can be viewed as the speed of travel of a given amount of data from node to another. In general, the greater the bandwidth of a given path, the higher the data transfer rate.

3.2.2 Packet Drop Rate

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. The fraction of lost packets increases as the traffic intensity increases. Therefore, performance at a node is often measured not only in terms of delay, but also in terms of the probability of packet loss. The packet loss rate is measured as the number of packets dropped in a simulation run.

3.2.3 End-to-end Delay

The End-to-end delay is measured as the time elapsed while a packet travels from one point e.g., source node to destination node. The larger the value of delay, the more difficult it is for transport layer protocols to maintain high bandwidths. We have calculated end-to-end delay by of the queue management algorithms and compared the results.

4. SIMULATION RESULTS AND ANALYSIS

In our simulation, we have compared the most popular queue management algorithms. We have studied the change in the Packet drop rate and end-to-end delay with varying data rate. Also, observed and analyzed a better algorithm in terms of the above mentioned metrics.

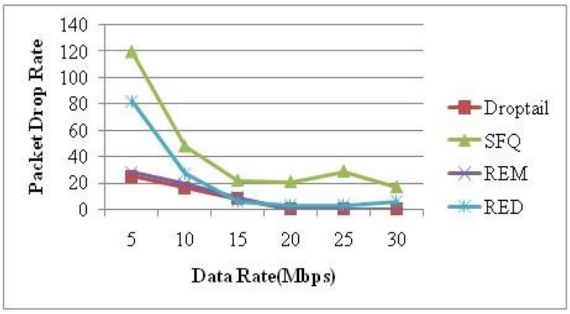


Figure 4: Analysis of Packet Drop Rate

4.1 Performance of Packet Drop Rate versus Data rate

Figure 4 shows the result of Packet drop rate versus data rate by running the topology shown in figure 2 and 3

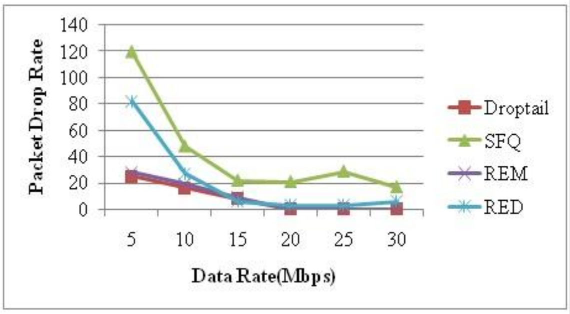


Figure 5: Analysis of Packet Drop Rate

It is observed that with the increase in data rate, the Packet Drop Rate gradually decreases. There was a smooth decrease in the Drop rate in REM, as REM did not show a higher Packet Drop Rate in the first place. In RED, as the data rate increased, the drop rate decreased and became steady at 20 Mbps comparable to Droptail and REM. But, SFQ increased its drop rate at 26 Mbps because of its unfairness. We also compared REM and SFQ as bottleneck link and droptail as other and the result is shown in figure 5.

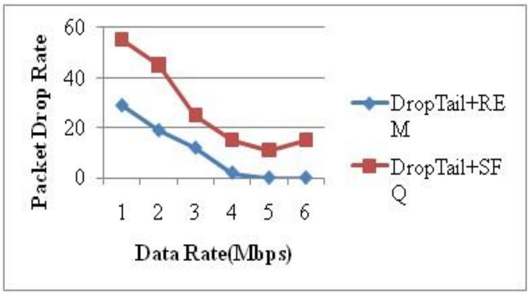


Figure 6: Analysis of Packet Drop Rate

It is observed that SFQ when used as bottleneck link results into less Packet drop rate.

4.2 Performance of End-to-end delay versus Data rate

As shown in figure 6, the End-to-end delay gradually decreases as the Data rate increases. Our results show that RED has less End-to-end delay as compared to others. In terms of delay, RED is a better queue management technique with lower data rate. But as gradually data rate increases to 30 Mbps, SFQ and REM performs equivalently for this topology

5. CONCLUSION

In this paper, we have made an effort to understand various popular queue management techniques and compare them using various parameters. Through this paper, we tried to understand the queue management techniques leverage the traffic loaded network. As our results show that, not a single algorithm is self sufficient. We have calculated Packet drop rate and end-to-end delay for the given topology. Our results show that REM has minimum packets dropped, while SFQ has the highest packets being dropped or lost in the network. While RED is an intermediate in terms of lost packets. But, simultaneously, if we consider end-to-end delay, RED achieved the best results. Our simulation results conclude that not a single queue management technique is sufficient in terms of all the parameters.

6. REFERENCES

[1] Johanna Antila, 51189d ,TLT "TCP Performance Simulations Using Ns2".

[2] T. Bonald, et al., "Analytic Evaluation of RED Performance", in Proceedings of IEEE INFOCOM, 2000.

[3] *Sanjeewa Athuraliya*. *Victor H. Li*. *Steven H. Low*. *Qinghe Yin*. Yin, "REM:Active Queue Management", in October 27, 2000.

[4] 1T.Bhaskar Reddy, Ali Ahammed , Reshma banu, "Performance Comparison of Active Queue Management Techniques" in February 2009.

[5] Sanjeev Patel, P. K. Gupta, Arjun Garg , Prateek Mehrotra and Manish Chhabra, in "Comparative Analysis of Congestion Control Algorithms Using ns-2", IJCSI, Vol 8, Issue 5, no.1 , September 2011.

[6] Kevin Fall, Editor Kannan Varadhan "The *ns* Manual" November 4,2011.

[7] Olawoyin, L.A, Faruk, N, 3Akanbi, L.A ", Queue Management In Network Performance Analysis".

[8] Jae Chung and Mark Claypool "Analysis of Active Queue Management" In *Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications (NCA)*, Cambridge, Massachusetts, USA April 2003.

[9] 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.

[10] Wu-chun Feng , Apu Kapadia , Sunil Thulasidasan, "Proactive Queue Management over a best-effort network", Proceedings of GLOBECOM 2002