International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 7, July 2014

Novel Approach for Queue Management and Improvisation of QOS for Communication Networks

J. Venkatesan[1], S. Thirumal[2]

[1]M. Phil, Research Scholar, Arignar Anna Govt. Arts College, Cheyyar, T.V Malai Dt, India

[2]Assistant Professor & Head, Department of Computer Science, Arignar Anna Govt. Arts College, Cheyyar, T.V Malai Dt, India

Abstract: ***Network resources are shared as needed by a community of users. Without effective traffic controls, networks are vulnerable to possible congestion when the offered traffic exceeds the network capacity, leading to serious deterioration of network performance. Network based congestion avoidance which involves managing the queues in the network devices is an integral part of any network. Most of the networks use Drop-Tail queue management where packets are dropped on queue overflow which is global synchronization problem. This packet loss results in increased overhead in terms of energy wasted to forward a packet which was dropped, additional energy required to retransmit this packet. Active Queue Management has been a solution to the global synchronization problem and it is a very active research area in networking flows. In order to stem the increasing packet loss rates caused by an exponential increase in network traffic, researchers have been considering the deployment of active queue management algorithms. Several AQM schemes were proposed to overcome congestion by effectively managing the queues. However, there are several drawbacks in each scheme and there is a need for AQM scheme that overcome all the drawbacks and effectively manages the queue thereby avoiding congestion. In this paper an Effective queue management strategy that manages the queue proactively and simple to implement is proposed and its performance is evaluated using ns-2.***

Keywords: QoS, Networks, Queue Management

1. Introduction

The Internet is essentially a network of interconnected queues. The two most fundamental experiences of a packet whilst traversing this network are delay and loss. The Internet is facing increasing packet loss rates and queuing delays. Lost packets waste resources, and may result in congestion collapse whereas queuing causes packet delay which reduces the quality of interactive applications. Active queue management algorithms were introduced to alleviate the problems of network congestion. In general, AQM schemes control congestion by controlling flow. Congestion is measured and a control action is taken. There are two approaches for measuring congestion: (1) queue based, and (2) flow based. In queue based AQMs congestion is observed by queue size. The drawback of this is that a backlog of packets is inherently necessitated by the control mechanism, as congestion is observed when the queue is already positive. This creates unnecessary delay and jitter. Flow based AQMs, on the other hand, determine congestion and take action based on the packet arrival rate. For such schemes, backlog, and all its adverse implications, is not necessary for the control mechanism. The total Network delay is essentially the sum of queuing delay and propagation delay. Currently queuing delay dominates most round trip times. The goal should be to reduce the network delay to just the propagation delay. Currently packet loss is both a signal of congestion and result of overflowing queues. The Internet engineering task force is introducing explicit congestion notification to feedback congestion by packet marking instead. With ECN, AQM queues can operate with minimal packet loss. The next generation of network paradigm will have AQMs that maintain high link utilization whilst achieving the QoS requirements of limited packet loss and delay. In this project, a new AQM scheme called Effective Queue Management that achieves this goal is proposed.

The aim is to design an Effective Queue Management for networks that avoids the congestion effectively than the existing schemes by overcoming their drawbacks and thereby improving the quality of service of it.

Even though several Active Queue Management algorithms already exist, none of them are without drawbacks. They have several deficiencies in effectively avoiding the congestion. If benefits in terms of improved performance are to be reaped by deploying AQM, it is imperative that any such scheme should be lightweight, Proactive, easy to implement and avoids the congestion by effectively managing the queues.

2. Literature Survey and Related Works

In this paper [1], Random Early Detection gateways for congestion avoidance in packet-switched networks is presented. RED is a first generation AQM. The rate of congestion notification is a function of the queue size. The gateway detects incipient congestion by computing the average queue size. The gateway could notify connections of congestion either by dropping packets arriving at the gateway or by setting a bit in packet headers. When the average queue size exceeds a preset threshold, the gateway drops or marks each arriving packet with a certain probability, where the exact probability is a function of the average queue size. RED gateways keep the average queue size low while allowing occasional bursts of packets in the queue. During congestion, the probability that the gateway notifies a particular connection to reduce its window is roughly proportional to that connection's share of the bandwidth through the gateway. RED gateways are designed to accompany a transport-layer congestion control protocol such as TCP. The RED gateway has no bias against bursty traffic and avoids the global synchronization of many connections decreasing their window at the same time. As discussed in depth in RED suffers from severe shortcomings. The queue size is not a good indicator of the severity of the congestion, and the level of congestion notifications issued may be too great and bursty, leading to excessive packet loss. RED is prone to periods of high loss followed by link under utilization.

In this technical report [2], they recommended AQM mechanism for effective communication. Queue management algorithms manage the length of packet queues by dropping packets when necessary or appropriate, The traditional technique for managing router queue lengths is known as "tail drop", since the packet that arrived most recently is dropped when the queue is full. This method has served the Internet well for years, but it has two important drawbacks. They are Lock-Out and Full Queues. AQM overcomes the above drawbacks and its advantages are like, it reduces the number of packets dropped in routers, it provides lower-delay interactive service, and avoids lock-out behavior. Internet routers should implement some active queue management mechanism to manage queue lengths, reduce end-to-end latency, reduce packet dropping, and avoid lock-out phenomena within the Internet. The default active queue management mechanism for managing queue lengths to meet these goals in FIFO queues is Random Early Detection. RED algorithm drops arriving packets probabilistically. The probability of drop increases as the estimated average queue size grows. It responds to a time-averaged queue length and not an instantaneous one. The RED algorithm itself consists of two main parts: estimation of the average queue size and the decision of whether or not to drop an incoming packet. RED effectively controls the average queue size while still accommodating bursts of packets without loss. Unless a developer has reasons to provide another equivalent mechanism, they recommend that RED be used.

BLUE [3] is a hybrid flow and queue based congestion control scheme. It uses packet loss (queue) and link under-utilization (flow) events to adjust the rate of congestion notification. The congestion notification rate pm is increased at a set rate if the queue size exceeds a threshold L, and it is decreased if the link is idle. The amount of increase is d1, and decrease d2, at a rate of 1/freezetime. We uncovered a subtle drawback with BLUE that may limit its practicability. BLUE behaves well if operated within a region of RTT and number of connections N for which the parameters d1 and d2 were set. However, changes in the dominant RTT of the connections going through the queue, or a significant change in N can invalidate the parameter settings and lead to queue backlog oscillation between loss and under-utilization. The amount of change of notification rate pm during a queue full or link idle event, is the product of the time spent in this event multiplied by the rate of change d(1,2)/freezetime. This time is related to the delay in the response of the TCP sources to the changed notification rate (2 x RTT). The greater the RTT, the greater will be the pm adjustment. If the RTT increases, so does the change in pm and this may result in backlog oscillation. This was observed in simulations using the recommended parameter settings of. Another cause of instability is a large change in the number of connections. It is not our intention to explore BLUE in depth here, but this instability is the result of the adjustment of congestion notification rate pm by a constant d1 or d2, despite the non linear relation of pm and N. Recall that based on the TCP Friendly Equation the function of pm versus N requires larger changes of pm for larger N.

This paper [4] describes a mechanism called SRED. Like RED, SRED pre-emptively discards packets with a load-dependent probability when a buffer in a router in the Internet or an Intranet seems congested. SRED has an additional feature that over a wide range of load levels helps it stabilize its buffer occupation at a level independent of the number of active connections. SRED does this by estimating the number of active connections or flows. This estimate is obtained without collecting or analyzing state information on individual flows. The same mechanism can be used to identify flows that may be misbehaving, i.e. are taking more than their fair share of bandwidth. Since the mechanism is statistical in nature, the next step must be to collect state information of the candidates for "misbehaving", and to analyze that information. We show that candidate flows thus identified indeed have a high posterior probability of taking a larger than average amount of bandwidth.

This paper [5] describes and evaluates Stochastic Fair Blue, a novel technique for enforcing fairness among a large number of flows. It protects TCP flows against non-responsive flows using the BLUE algorithm. SFB scalably detects and rate-limits non-responsive flows through the use of a marking probability derived from the BLUE queue management algorithm and a Bloom filter. SFB is highly scalable and enforces fairness using an extremely small amount of state and a small amount of buffer space. SFB is based on two independent algorithms. The first is the BLUE queue management algorithm. This algorithm uses a single marking probability to mark packets in times of congestion. The heavier the congestion is, the higher the marking probability. The second algorithm is based on Bloom filters. This algorithm allows for the unique classification of objects through the use of multiple, independent hash functions. Using Bloom filters, object classification can be done with an extremely small amount of state information. Using analysis and simulation, SFB is shown to effectively handle non-responsive flows using an extremely small amount of state information.

In this paper [6], a global architecture for Internet host distance estimation and distribution which is called IDMaps, Internet Distance Map Service is proposed. Its intended IDMaps be the underlying service that provides the distance information used by SONAR/HOPS. It measures and disseminates distance information on the global Internet. Higher-level services can collect such

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 7, July 2014

distance information to build a virtual distance map of the Internet and estimate the distance between any pair of IP addresses. The goal is to provide distance information in terms of latency and, where possible, bandwidth. In the context of nearest mirror selection for clients, we showed that significant improvement over random selection can be achieved using placement heuristics that do not require a full knowledge of the underlying topology. In addition, we showed that IDMaps overhead can be minimized by grouping Internet addresses into APs to reduce the number of measurements, the number of Tracers required to provide useful distance estimations is rather small, and applying t-spanner to the Tracer-Tracer VLs can result in linear measurement overhead with respect to the number of Tracers in the common case. Overall, this study has provided positive results to demonstrate that a useful Internet distance map service can indeed be built scalably. Through Internet experiments and simulations, it is showed that this approach can indeed provide useful distance information.

In this paper [7], a proactive queue management scheme called GREEN, which regulates TCP flows over the same link to a fair sending rate and hence prevents them from inducing congestion, is proposed. It does so by using the knowledge of TCP's steady state behavior. It exhibits high fairness with flows of widely varying RTT's. However their design suffers from severe under utilization in the presence of short lived or low bandwidth flows. From the detailed study of the previous works, it is clear that the Active Queue Management schemes such as RED, BLUE, SRED, REM do not provide adequate fairness at the cost of higher utilization and for widely varying RTT's. They also suffer from severe under utilization in the presence of short lived or low bandwidth flows.

3. Existing System

Most of the networks use Drop-Tail queue management where packets are dropped on queue overflow which is global synchronization problem. Random Early Detection was proposed as a solution to the 'Global Synchronization' problem and this opened up a new area of research called Active Queue Management. The key aims of AQM are to prevent global synchronization, reduce queuing delays and improve resource utilization.

4. Proposed System

The proposed system is lightweight, proactive and hence is ideally suited for deployment. It reduces Packet loss ratio, Increases transmission efficiency, Computational overhead is negligible. It avoids the congestion by effectively managing the queues.

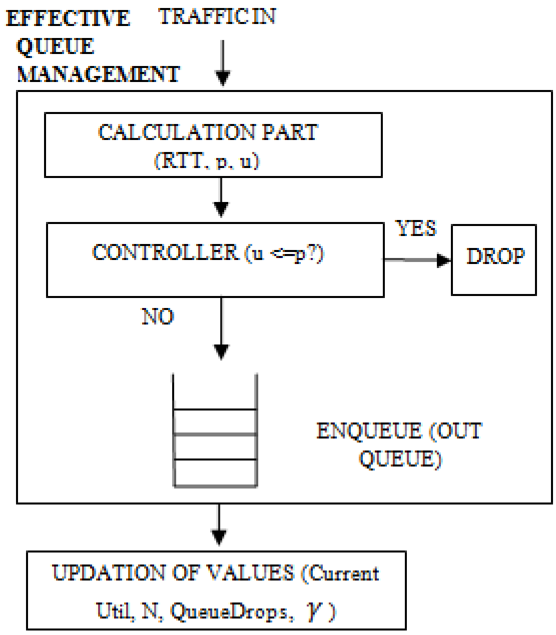

Figure 1: Effective Queue Management Architecture

4.1 System Architecture

When a packet is received at the EQM router, EQM first obtains the packet's RTT. Then it calculates the value of p (packet drop probability) and u (random number selected over a uniformly – distributed interval [0, 1]). If u <= p then the packet is dropped else it is added to the outgoing queue. If the window of time has elapsed for updating the values of current link utilization (currentUtil), the number of active flows (N), and the number of queue drops due to overflow (queueDrops), then these values are updated. Additionally the value of γ is adjusted based on currentUtil and queueDrops.

4.2 Module Description

4.2.1 Traffic IN

It signifies incoming of the packets from the sender into the EQM router.

4.2.2 Calculation Part

Calculation of RTT: When a packet enters, its RTT (Round Trip Time) is calculated. The method which is used for calculation of RTT is Embedded RTT's. It requires TCP senders to embed their current RTT estimates within the TCP header.

Calculation of p: It Signifies Packet Drop Probability. It is calculated using the Formula,

$$p = \left(\frac{N \times MSS \times c}{\gamma(t) \times L \times RTT} \right)^2$$

Where,
N → No of Active Flows
L → Outgoing Link capacity
MSS → Maximum Segment Size
RTT → Round Trip Time
c → A Constant value that depends on the acknowledgement strategy used

$\gamma(t)$ → A Constant at time "t"

Calculation of u: It's a random number selected over a uniformly-distributed interval [0, 1].

4.2.3 Controller

If u<=p then the packet is dropped (i.e., each packet is probabilistically dropped with the calculated probability p) and otherwise it is added to the outgoing queue.

4.2.4 Updation of Values

If the window of time has elapsed for updating the values of current link utilization CurrentUtil, the number of active flows N, and the number of queue drops due to queue overflow QueueDrops, then these values are updated. Additionally, the value of γ is adjusted based on CurrentUtil and QueueDrops.

5. Simulation SETUP

5.1 Pseudo Code for EQM Algorithm

```
Enqueue(Packet pkt)
RTT ← obtainRTT(pkt)
P ← [(N*MSS*c)/(γ*L*RTT)]^2
u ← UniformRand(0,1)
if (u<=p) then
        drop(pkt)
else
        addToQueue(pkt)
endif
if (currentTime() – lastUpdate >= window) then
        update(currentUtil, N, queueDrops)
        lastUpdate ← currentTime()
                if(queueDrops > 0 ) then
                        γ ← 0.95γ
                elseif currentUtil < 0.98 then
                        γ ←
                [(1+currentUtil)/(2*currentUtil)] γ
                end if
end if
```

5.1.1 EQM-Effective Queue Management

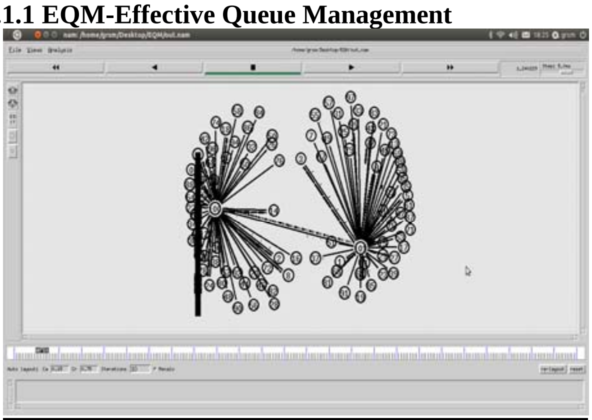

Figure 5.1: EQM Nam Window

5.1.2 End to End Delay - XGraph

End-to-end delay refers to the time taken for a packet to be transmitted across a network from source to destination. The End to end delay for the existing system is plotted using GNU plot.

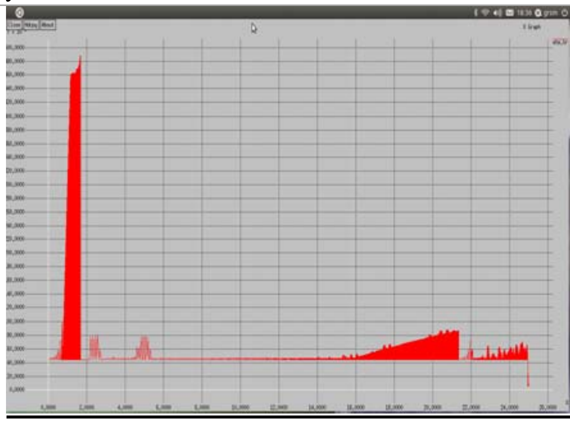

Figure 5.2: EQM End To End Delay

6. Performance Analysis

The necessary inputs are given to the Existing and Proposed systems and the output is received. From those output parameters, using Trace File, AWK commands and gnuplot, XGraphs and Bargraphs are drawn. Then using these graphs the performance is evaluated.

6.1 Average Queue Size

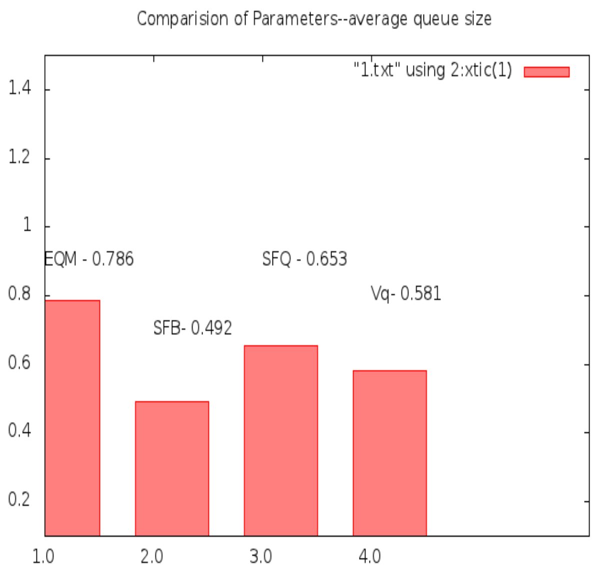

Figure 6.1: Average Queue Size

The Average Queue Size of EQM is higher than the existing schemes such as SFB, SFQ, Virtual queue etc.

ISSN (Online): 2347-3878
Volume 2 Issue 7, July 2014

6.2 Packet Drop

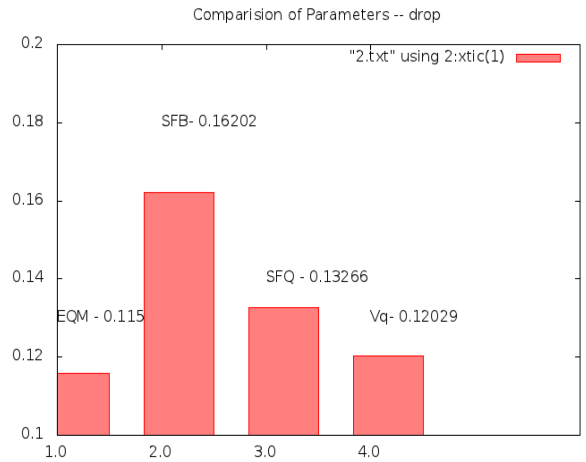

Figure 6.2: Packet Drop

The Packet Drop in EQM is lower than the existing schemes such as SFB, SFQ, Virtual queue etc..

6.3 Fairness

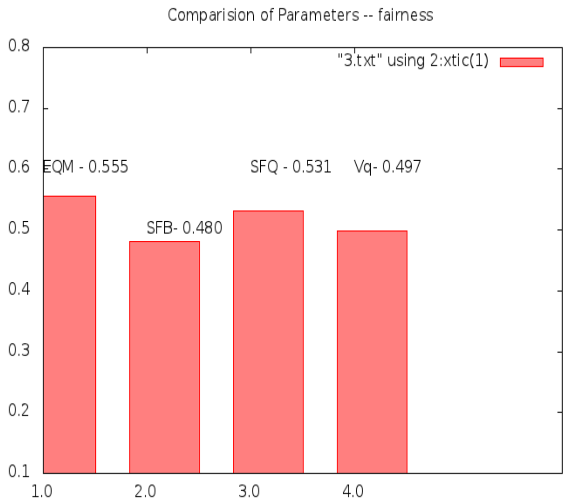

Figure 6.3: Fairness

The Fairness of EQM is higher than the existing schemes such as SFB, SFQ, Virtual queue etc.

6.4 Link Packet Loss

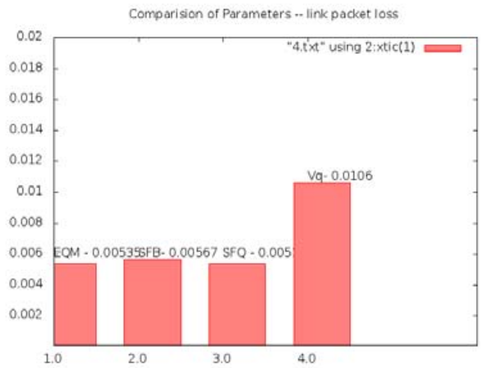

Figure 6.4: Link Packet Loss

The Link Packet Loss of EQM is Lower than the existing schemes such as SFB, SFQ, Virtual queue etc..

Fig 5.14 Throughput

The Throughput of EQM is higher than the existing schemes such as SFB, SFQ, Virtual queue etc.

6.5 Utilization

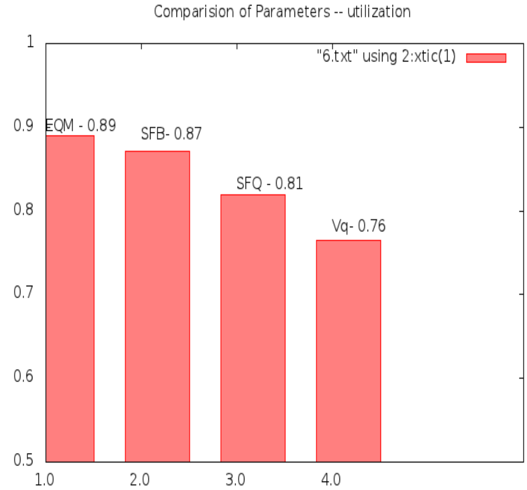

Figure 5.15: Utilization

The Utilization of EQM is higher than the existing schemes such as SFB, SFQ, Virtual queue etc.

7. Conclusion and Future Work

In this paper a new scheme called Effective Queue Management that overcomes the deficiencies in well known queue management algorithms has been proposed. Its effectiveness over the existing algorithms is proved using the comparison graphs. The algorithm is simple, robust, low in computational complexity, easily configured, and self-contained to a single router, making it easy to deploy. Deployment of low delay, low loss algorithm such as EQM will improve Internet performance and enable real-time applications. In future effectiveness of EQM will be analyzed using various parameters and further improvements will be done for various scenarios. It will also be deployed in MANET and its effectiveness will be analyzed.

References

[1] S. Floyd and V. Jacobson "Random early detection gateways for congestion avoidance". IEEE/ACM Transactions on Networking, 1(4):397–413, Aug. 1993.

[2] R. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V.J.G. Minshall, C. Patridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. "Rfc- 2309 recommendations on queue management and congestion avoidance in the internet". Technical report, IETF, pp 246–350, 1998.

[3] W. Feng, D. Kandlur, D. Saha, K. Shin, "Blue: A New Class of Active Queue Management Algorithms" U. Michigan CSE-TR-387-99, April 1999.

[4] T.J. Ott, T.V. Lakshman, and L.H. Wong. "SRED: Stabilized RED. In IEEE INFOCOM", volume 3, pages 1346– 355, 1999.

[5] W. Feng, D. Kandlur, D. Saha, K. Shin, "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness", in Proc. of INFOCOM 2001, April 2001.
[6] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: A Global Internet Host Distance Estimation Service", IEEE/ACM Transactions on Networking, 2001 October.
[7] W.Feng, A.Kapadia and S.Thulasidasan. "GREEN: Proactive Queue Management over a Best-Effort Network". in proc of IEEE Globecom 2002, November 2002.
[8] The Network Simulator - ns-2 homepage: http://www.isi.edu/nsnam/ns/