# Fairness and Percentage Loss Analysis for Short Lived Web Transfers using QMBCCA

V. Sinthu Janita
Asst Professor & Head
Dept of Computer Science
Cauvery College for Women
Tiruchirappalli, India.

Dr. D.I.George Amalarethinam
Director - MCA
Jamal Mohammed College
Tiruchirappalli, India.

Dr.E.George Dharma PrakashRaj
Asst Professor in Computer Science
Bharathidasan University
Tiruchirappalli, India.

## ABSTRACT
One of the major problems in Internet today is Network Congestion. TCP is used as a Transport Protocol for most of the applications on the Internet and is also responsible for adjusting to network congestion dynamically. The Active Queue Management algorithms (AQM) are used to reduce congestion, and in this paper two such AQM algorithms are considered. Here we look at the existing Explicit Congestion Notification (ECN) algorithm and a Queue Management Backward Congestion Control Algorithm (QMBCCA). A comparison is made between the two algorithms in terms of Fairness and Percentage Loss for short lived web transfers. We have found that there is a significant reduction in packet loss and improvement in fairness index when using QMBCCA when there were more number of TCP flows.

## Key Words
Congestion Control, Active Queue Management, RED, ECN, Packet Loss, Fairness

## 1. INTRODUCTION
Congestion occurs when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs [1]. The normal method that a router reacts to congestion is by dropping the incoming packets in the absence of buffer space. This is commonly called as "Tail Drop" or "Drop Tail". The dropping of incoming packets is an indication to the transport layer that there is congestion in the network.

TCP is the most leading transport protocol today [2]. Whenever congestion occurs in the network, an intermediate router begins to drop packets. In response to this hint of congestion, TCP invokes its congestion control algorithms [3] so that congestion is eased. These congestion control algorithms make TCP responsive even to a single packet loss. The use of more web applications which use TCP as their transport protocol has resulted in an increase in network traffic, which leads to, increased network congestion and increased packet drops. This affects the performance of TCP drastically. With the increase in the amount of traffic and congestion in the Internet, it is no longer practical to rely only on the congestion control algorithms to provide good service at al1 times.

Many research works have gone into how the network can involve itself to complement these congestion control mechanisms. This is where the Queue Management algorithms are used "Queue Management" algorithms deal with managing the lengths of packet queues by dropping packets whenever necessary. The main concept of these algorithms is to involve the routers for easing congestion before it actually occurs in the network. Dropping packets probabilistically even before the queue fills up in the router eases congestion. This makes the sender TCP to back off and ease congestion before it actually occurs. Random Early Detection (RED) is one of the queue related algorithms that is popular in recent years. RED [4] follows the concept of Early Random Drop by introducing the measure of average queue size and also dynamically changing the drop probability.

The algorithm, called Loss Ratio-based RED (LRED), measures the latest packet loss ratio, and uses it as a complement to queue length for adaptively adjusting the packet drop probability. This achieves fast response time and yet good robustness.[5] Explicit Congestion Notification (ECN) algorithm is an extension of the RED algorithm that marks a packet instead of dropping it. [6] proposes a new AQM algorithm that exploits Round Trip Time explicitly by introducing a passive RTT estimation technique in a router.

PERED (pre-Estimation RED) is a new active queue management algorithm which creates a dynamical Markov model according to the traffic loads to estimate the future queue size of router buffer and self-adjusts the parameters of well-known Random Early Detection (RED) to get better performance[7].

RED requires careful tuning of its parameters in order to provide good performance. A new mechanism BO-ARED (An Adaptive RED Algorithm Combined With Buffer Occupation) improves the performance of RED by matching router's buffer occupation with $w_q$, $min_{th}$, $max_{th}$ and $P_{max}$ parameter settings, to make BO-ARED adapt to network environment variation automatically[8].

The QMBCCA algorithm is the algorithm that has been proposed by us [9]. QMBCCA work in combination with RED and ECN to provide congestion feedback to the sender without actually dropping the packets. The advantage is that it reduces the number of packet drops, which is expected to benefit TCP during times of congestion.

The paper is organised as follows. Section 2 gives an introduction on the Active Queue Management (AQM) Algorithms, specifically explaining about the RED and ECN Algorithms. Section 3 explains the QMBCCA algorithm. The Simulation Screenshots and the corresponding graphs are given for the two parameters chosen for comparing the two algorithms in Section 4. Section 5 concludes the paper.

## 2. AQM ALGORITHMS
Active Queue Management algorithms basically detect congestion before the queue overflows and it also gives an indication of the congestion that has occurred to the end nodes [4]. The transport protocols do not have to rely on buffer overflow as the only indication of congestion in an active queue management. This is because they get notified before congestion happens. The active queue management allows routers to control and count how many

packets are to be dropped by dropping packets before the buffers get overflowed.

The following section describes the RED, which is an active queue management algorithm and its extension algorithm called as ECN.

## 2.1 Random Early Detection

Random Early Detection [4] follows the concept of Early Random Drop by introducing the measure of average queue size and also dynamically changing the drop probability.

The use of dynamic drop probability ensures that the gateway reacts differently to different level of congestion. Also the instantaneous queue sizes are not a correct indicator of congestion due to burst nature of Internet traffic. RED therefore uses average queue size measured over all times. The RED gateway finds the average queue size as exponentially weighted moving average of instantaneous queue size. The minimum threshold and the maximum threshold are compared with the average queue size.

Every arriving packet is marked when the average queue size is between the maximum threshold and minimum threshold. When the mechanism ensures that the marked packets are dropped, or if all source nodes are co-operative, then the average queue size does not exceed the maximum threshold.

## 2.2 Explicit Congestion Notification

Explicit Congestion Notification (ECN) is an extension proposed to RED that marks a packet instead of dropping it [10]. The packets are sent once it is marked to the recipient who will in turn inform the sender about the just beginning congestion. This will in turn trigger congestion avoidance algorithm by the sender to ease congestion. The advantage of using ECN is that, it not only stops the packets being dropped, but it also avoids congestion in the router. ECN requires support from both the router as well as the end hosts. The packet from flows that are not ECN capable will continue to be dropped by RED.

The ECN algorithm has the following drawbacks.

1.ECN is attached to the transport layer through the use of header information. This scheme can be extended to other transport protocols and will require changes to each of their respective headers.

2.ECN requires the congestion notification to incur a Round Trip Time (RTT) before the sender can react. In a path with high delay bandwidth, this would create two problems.

The First problem is caused when the delay bandwidth product is dominated by a high bandwidth, and when a large amount of traffic passes through the intermediate routers. This will lead to an increase in congestion level before the sender is notified.

The second problem is when the delay bandwidth product is dominated by the high latency then the reaction will take too long to address the congestion issue. In both the situations, the efficient use of the available bandwidth is affected. These drawbacks are rectified in QMBCCA[9].

# 3. QMBCCA

Queue Management Backward Congestion Control Algorithm (QMBCCA) is an alternative approach to the current ECN algorithm involving the use of an existing Internet Control Messaging Protocol (ICMP) Source Quench (ISQ) signaling mechanism [9]. Congestion notification is kept at the IP level.

The use of ISQ based approach reduces the reaction time to congestion in the network. Also, a mechanism is proposed in which the ISQ message can include information on the severity of the congestion allowing the end host to react accordingly and adjust the window so as to make maximal use of the resources and thereby maximizing network utilization.

A connection is notified by an ISQ of congestion at a rate proportional to the connection's share of the bandwidth at the congested router. The generation of ISQ messages will be limited to the period between the initial congestion detection and until the source end system adjusts.

Given this scheme which addresses congested router sequentially on a forward path, it can be assumed that the back path is probably not really congested since it covers the path only to the first point of congestion along that path.

It can be said that when using RED, less packet drops happen at the router in comparison to the traditional drop tail algorithms. This implies the amount of processing needed at the router is reduced. It can be said that RED addresses both the backward path congestion problems, if the back path is the same one as the forward path as well as the router processing concerns.

Both the CE (Congestion Experienced) and ECT (ECN Capable Transport) bits are maintained as in ECN [9]. When the IP message is demultiplexed, the values of both CE and ECT are passed to the transport layer. For packets with the ECT bit set in the IP header, the CE bit in the header set is forwarded instead of the packet being dropped. These packets are marked with a given probability if the average queue size lies between the minimum and maximum thresholds. ISQs are generated by the intermediate congested RED router and sent back to the source as an indication of initial congestion whenever that router decides to mark the CE bit.

ISQs are usually not generated for a packet that has been marked previously by another router regardless of whether that packet is contributing to some congestion. However when the router queue level informs that the packet be dropped, then an ISQ is sent back to the source regardless of whether the packet has been marked previously or not. The source reacts at the transport protocol level by lowering its data throughput into the network. In TCP, upon identifying the flow causing the congestion, the sender reacts by halving both the congestion window and the slow threshold value for that flow.

A QMBCCA capable router is assumed to have an active queue management algorithm implemented in it. Since RED is considered as the base algorithm, some modifications on the routers have to be made.

The First modification is that if the incoming packet causes the average queue size to go above the maximum threshold, then the packet is dropped. If the ECT bit is set in the IP header then, send an ISQ back to the source.

The Second modification is that if the incoming packet causes the average queue to go between the minimum and maximum thresholds, RED probability may choose this packet to be marked. If the ECT bit of this packet is set and if it is not already marked (i.e. the CE bit is not set) then mark the packet (i.e. set the CE bit) and send back an ISQ. If the ECT bit is not set and RED chooses this packet, then the packet is dropped. The algorithm for QMBCCA is given below. The flowchart of QMBCCA is in [9].

For each packet arrival

calculate the average queue size *avg*

if $min_{th} <= avg < max_{th}$

```
{
    Calculate probability pa
    With probability pa:
     if ECT bit is set and if the packet
         is not  already marked
            {
              Mark the arriving packet
              Send an ISQ to the sender
            }
     else
              Drop the arriving packet
  }
  else if max_th <= avg
              Drop the arriving packet
Send an ISQ to the source.
```

## 4. SIMULATIONS AND ANALYSIS

The study and comparative evaluation of ECN and QMBCCA mechanisms is done using Java Programming [7]. In this section, we describe the simulation parameters and the performance metrics used in evaluating the algorithms. Network topology chosen for our experimentation is given in Figure 1.

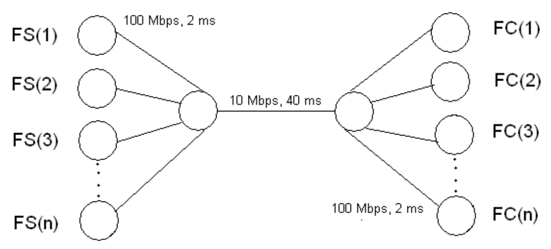


**Figure 1. Network Topology**

This experimentation assesses the performance of QMBCCA with short-lived web-traffic workloads. The parameters chosen for experimentation are as follows

1) Fairness Index for web object transfer delay*:* For N transferred objects; the Fairness Index (FI) for transfer delay is computed using the following formula:

$$FI= \frac{\left(\sum_{i=1}^{N} T_i\right)^2}{\left(\sum_{i=1}^{N} T_i{}^2\right)}$$

where Ti is the transfer delay incurred by object i. This is the interval between the time a web client makes an initial request and the time the server receives the ACK to the last data packet for the object requested by client.

2) Percentage loss: Measures the ratio of the number of packets dropped at the bottleneck link to the total number of packets injected into the bottleneck link for a particular flow or set of flows.

In the experiment, the web sessions are set up between 10 web servers and 20 web clients, while FTP connections are varied using 5, 10, 15, 20, 25, 30, and 35 flows, to establish different levels of congestion. The FTP flows start randomly first followed by the web-traffic. Short-lived traffic sources were simulated for 1000 sessions with constant session size of 10 pages, a minimum of 3 objects in each page. The inter page time was 10s and inter object time was 0.1s. The size of an HTTP object is taken as 10KB. Fairness index for object transfer time was calculated since all the objects had the same size. Several sessions were active throughout simulation time. Fairness in web object transfer times and the percentage loss for HTTP packets were recorded.

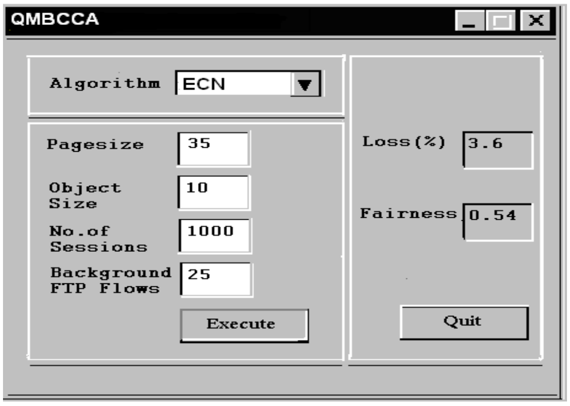


**Figure 2. ECN Screenshot for Background Flows = 25**

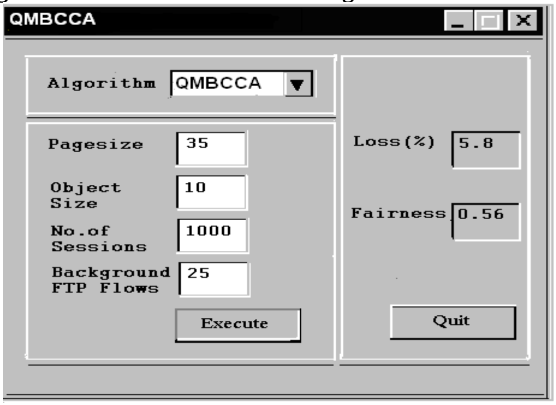


**Figure 3. QMBCCA Screenshot for Background Flows = 25**

Figure.2 and Figure.3 displays the screenshot that calculates the Loss Percentage and Fairness with respect to the ECN and QMBCCA Algorithms.

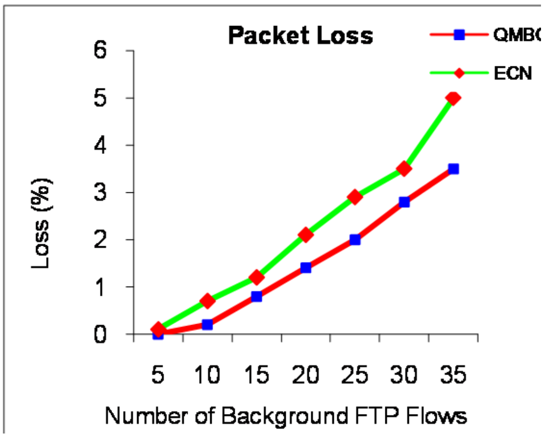


**Figure 4. Packet Loss**

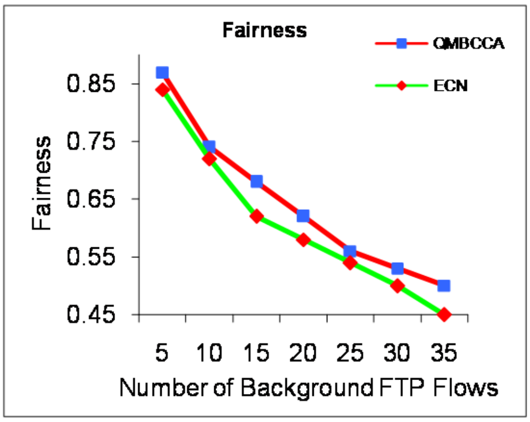


**Figure 5. Fairness**

Figure.4 and Figure.5 show the observed results alongside results for ECN. Figure 4 shows that even though the burst ness of the web-traffic workloads causes packet loss, QMBCCA experiences around 40% less packet loss than ECN connections. QMBCCA has small mean queue size with less queue fluctuations compared to the ECN and hence the less packet loss. Figure 5 shows around 15% improved fairness index for QMBCCA

## 5. CONCLUSION

This paper explains about the existing Explicit Congestion Notification (ECN) algorithm and the Queue Management Backward Congestion Control Algorithm (QMBCCA). Both the algorithms are based on the Random Early Detection (RED) algorithm. A comparison of the ECN and QMBCCA algorithms in terms of Fairness and Packet loss is done. We have found that there is a significant reduction in packet loss and improved fairness index when using QMBCCA when there were more number of TCP flows. This work is done for short-lived web transfers and this can be extended for long-lived web transfers also.

## 6. REFERENCES

[1] Kevin Curran, Derek Woods, Nadene Mc Dermot and Colleen Bradley, "The Effects Of Badly Behaved Routers On Internet Congestion", International Journal Of Network Management Int. J. Network Mgmt, 13: 83 – 94, 2003.

[2] S. Mascolo, C. Casetti, M. Gerla, S. S. Lee, M. Sanadidi "TCP Westwood: congestion control with faster recovery" UCLA CSD Technical Report #200017, 2000.

[3] Seungwan Ryu, Christopher Rump, And Chunming Qiao, "Advances In Internet Congestion Control", Surveys, IEEE Communications, Vol 5, No.1, 2003.

[4] Serhat Özekes, "Evaluation Of Active Queue Management Algorithms" , *I*stanbul Commerce University Journal of Science, Vol.4, No.7, pp.123- 139, 2005.

[5] Chonggang Wang, Jiangchuan Liu, Bo Li, Kasem Sohraby, Y. Thomas Hou, "LRED: A Robust Responsive AQM Algorithm Using Packet Loss Ratio Measurement", IEEE Transactions on Parallel and and Distributed Systems, (vol. 18, no. 1), pp 29-43, Jan 2007

[6] H. Hoshihara, H. Koga, T. Watanabe, "A New Stable AQM Algorithm Exploiting RTT Estimation", Proceedings of the 31st IEEE Conference on Local Computer Networks, pp 143 -150,. 2006

[7] Cao Zhen-zhen; Xiao Yang, " PERED: A new AQM Algorithm", Proceedings of International Conference on Wireless, Mobile and Multimedia Networks, China, pp 1-4 China, August 2009

[8] Yunqi Luo Hong Chen Xiaoping Yang Yuxiang Hao "BO-ARED: A new AQM algorithm with adaptive Adjustment of parameters", Proceedings of 2010 8th World Congress on Intelligent Control and Automation (WCICA), China, pp 1852 – 1857, August 2010

[9] Sinthu Janita Prakash, D. I. George Amalarethinam, E. George Dharma Prakash Raj, Prasanna Ranjith " A Queue Management Backward Congestion Control Algorithm (QMBCCA)", Third International Conference of Basic Science, University of Al-Jabal Al- Gharibi,Libya pp 25- 27, 2009.

[10] J. Hadi Salim , U.Ahmed "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks", Request for Comments: 2884, 2000.

[11] Java - Programmer's Reference, Herbert Schildt with Joe O'Neil, Tata McGraw Hill