A Modification to RED AQM for CIOQ Switches

Jay Kumar Sundararajan, Fang Zhao, Pamela Youssef-Massaad, Muriel Médard
{jaykumar, zhaof, pmassaad, medard}@mit.edu
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139.

***Abstract*— In very large networks with heavy traffic, congestion control plays an important role in network resource management. One approach to this is the Active Queue Management (AQM) algorithms. Many AQM algorithms have been proposed and analyzed but they mainly focus on single queued links. Recognizing the fact that input queued switches are limited in throughput and output queued switches require a large speedup factor, we direct our attention to Combined Input and Output Queued(CIOQ) switches. We propose a simple modification to the RED AQM algorithm in order to account for the presence of both input and output queues in the switch. Specifically we use the weighted sum of input and output queue lengths as the congestion measure instead of just the output queue length. Simulations show that with such a simple modification, the average backlog in the switch is significantly reduced in the low speedup region as compared to RED without this modification. Unlike the traditional dynamic of having the loss rate grow with the length of the queue, simulations show that for a loss rate in the modified RED slightly larger than that in RED, the output queue length in modified RED is tremendously reduced. The weighting factor used in the computation of the congestion measure provides a means to balance the reduction in the average backlog on the one hand, and the increase in the loss rate on the other hand. Finally, simulations show that the improvement gained in terms of the queue length does not compromise in any way the utilization of the switch as compared to RED and Droptail.**

I. Introduction

Consider a network where sources compete for bandwidth and buffer space while being unaware of current state of resources and unaware of each other. In this setting, congestion arises when the demand for bandwidth exceeds the available link capacity. This leads to performance degradation in the network as packet losses increase and link utilization decreases. To avoid these problems, some kind of organization and control of traffic is needed.

One way to perform congestion control is to use Active Queue Management (AQM) in the routers. The basic idea behind an AQM algorithm is to sense the congestion level within the network and inform the packet sources about this so that they reduce their sending rate. Many AQM algorithms have been proposed so far, such as RED [7], GREEN [5], REM [1] and BLUE [6]. In particular, there have been several modifications to RED. Some references in this area include Dynamic RED (DRED) [9], Adaptive RED [8], Stabilized RED (SRED) [11], Jacobson et al. [10] and many others. However, these approaches mainly focus on output queued switches. Most practical routers have input queues also, as this would help reduce the required speedup in the switch.

The implementation of flow-based AQM algorithms on combined input output queued (CIOQ) switches has been studied in [4] through simulations. The paper uses the GREEN algorithm; the rates used as input to the algorithm incorporate, in addition to the output queue arrival rates, the rates of the input side VOQs. This idea is important since in a CIOQ switch, congestion affects input queues as well as output queues.

In this paper, we look at the impact of input queues on the design of queue-based AQM algorithms. Specifically, we propose a modification of the Random Early Detection (RED) algorithm to account for the presence of the input queue. Instead of using the output queue length as a congestion measure, we use a weighted sum of the input and output queue occupancies. The key advantage of this is that the RED (output) queue need not get filled up as much as in original RED to start reacting to congestion. To study the effect of this change, simulations were performed in network simulator (ns) for a 4×4 and a 16×16 CIOQ switch with virtual output queues (VOQs). Results show that our fairly simple modification of the RED algorithm significantly reduces the backlog hence reducing the packet queuing delay. Simulations prove that the utilization remains unaffected as compared to the original RED. The weighting factor given to the input queue length in the congestion measure is a critical parameter. We can use it to strike a balance between reducing the average backlog and preventing the loss rate from becoming too high.

The remaining part of this paper is organized as follows. Sections 2 and 3 give a brief review on AQM and switching. Section 4 discusses the effect of input queues on AQM design, and describes our proposed modification to RED. Simulation settings and results are presented in Section 5, and finally, Section 6 gives directions for future work.

II. AQM Background

This section describes the various types of congestion control mechanisms usually employed in the Internet with emphasis on AQM.

A. Congestion Control

The objective of congestion control is to achieve efficiency i.e. maximum utilization, minimum queue size and minimum packet drops while giving fair access to all the sources. There are mainly two approaches to congestion control – end system congestion control and network-centric congestion control.

In end system congestion control, senders detect the congestion and react to it accordingly. TCP is an important example of this approach. When a packet is dropped, the sender assumes that congestion has occurred and reduces the sending rate. When a packet is successfully transmitted, senders increase their rate.

The other approach is network-centric congestion control. The idea behind this is that since routers have more information about the state of the network, they can be useful in detecting congestion and should take part in the decision of congestion control. Routers actually measure the congestion level by comparing input traffic to capacity and by looking at the queue size; thus, they can send feedback as soon as they notice that the queue length is growing. Therefore, the average queue length doesn't have to be as large as in the previous approach. Routers could also be used to give priorities to some sources as compared to others. An important example of network-centric congestion control is Active Queue Management.

B. Active Queue Management

AQM refers to a class of algorithms designed to provide improved queuing mechanisms for routers. These schemes are called active because they dynamically signal congestion to sources even before the queue overflows; either explicitly, by marking packets (e.g. Explicit Congestion Notification) or implicitly, by dropping packets. There are two design considerations in any AQM - first, how the congestion in the network is measured and next how this measure is used to compute the probability of dropping packets [1].

Queue based AQMs couple congestion notification rate to queue size. The AQMs currently employed on the Internet, such as Droptail and RED belong to this category. Another example of a queue based AQM is BLUE [6]. The drawback of this is that a backlog of packets is inherently necessitated by the control mechanism, as congestion is observed only when the queue length is 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. Some examples are REM [1] and GREEN [5]. For such schemes, the target utilization can be achieved irrespective of backlog. In this project, we focus on queue based AQMs, Droptail and RED, which will be explained in more detail below.

1) ***Droptail****:* Droptail is the simplest AQM algorithm, and is most widely used in the Internet. Under Droptail, when queue overflow occurs, arriving packets are dropped. Although it is simple to implement, and has been tested and used for many years, it was shown to interact badly with TCP's congestion control mechanisms and to lead to poor performance. Congested links have high queue size and high loss rate.

2) ***Random Early Detection - RED****:* RED [7] has the potential to overcome some of the problems discovered in Drop-Tail such as synchronization of TCP flows and correlation of the drop events (multiple packets being dropped in sequence) within a TCP flow. Packets are randomly dropped before the buffer is full, and the drop probability increases with the average queue size.

RED is also a queue based AQM mechanism. RED gateways require the user to specify five parameters: the maximum buffer size or queue limit (QL), the minimum (min_{th}) and maximum (max_{th}) thresholds of the "RED region", the maximum dropping probability (max_p), and the weight factor (w_q) used to calculate the average queue size.

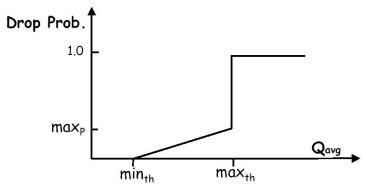

Fig. 1. Dropping Probability of RED AQM

It uses early packet dropping in an attempt to control the congestion level, limit queuing delays, and avoid buffer overflows. Fig. 1 shows the dropping probability of RED AQM as a function of the average queue length. Early packet dropping starts when the average queue size exceeds min_{th}, shown in the figure. The average queue size (Q_{avg}) is calculated as an exponentially weighted moving average using the following formula:

$$Q_{avg}(i) = (1 - w_q) \times Q_{avg}(i-1) + w_q \times q_{inst}, \quad (1)$$

where q_{inst} is the instantaneous queue size.

This weighted moving average captures the notion of long-lived congestion better than the instantaneous queue size. Had the instantaneous queue size been used as the congestion metric, short-lived traffic spikes would lead to early packet drops. So a rather under-utilized router that receives a burst of packets can be deemed congested if one uses the instantaneous queue size. The average queue size, on the other hand, acts as a low pass filter that allows spikes to go through the router without forcing any packet drops (unless, of course, the burst is larger than the queue limit). The user can configure w_q and min_{th} so that a RED router does not allow short-lived congestion to continue uninterrupted for more than a predetermined amount of time. This functionality allows RED to maintain high throughput and keep per-packet delays low.

Next, we will give a brief introduction about switching in routers to show the motivation behind CIOQ switches.

III. Switching Survey

Output-Queued Switches: An output-queued (OQ) switch is one in which only output links have buffers. Whenever a packet arrives at any input, it is immediately routed to the destination of the packet. Thus, no packet has to wait on the input side. For this switch to work, the switching fabric must have a speedup of n, the number of inputs (or outputs) of the switch. Speedup is the relative speed of the switching fabric compared to the speed of the input or output links. This strategy is known to maximize the throughput of the switch, as long as no input or output is over-subscribed. However, the main drawback is that this switch architecture is not scalable.

Input-Queued Switches: An input-queued (IQ) switch is one which has buffers only on the input side. The input buffers can hold the packets which couldn't be processed due to contention. Consequently, speedup required in the switch fabric is low and the switch architecture is scalable and can work for the high line rate case also. However, the main

drawback with the IQ switches is *head-of-line blocking*. It limits the maximum throughput of the switch to 58.6 % [2].

Virtual Output Queues: The problem of HOL blocking can be circumvented by the concept of virtual output queues (VOQs). Here, each input maintains a separate input queue for each output. The only problem is that the number of queues on the input side will scale with the square of the number of inputs or outputs. This is the price paid for the enhancement in throughput.

Combined Input Output Queued Switches: This is the most general type of switch. As the name suggests, there are buffers both on the input side and output side. The speedup of the switch can be anything between 1 and n. It has been shown that a CIOQ switch with a speedup of 2 can emulate an output-queued switch, and therefore, achieve 100 % throughput. [3] To avoid the problem of HOL blocking, it is possible to have a CIOQ switch with virtual output queues on the input side . Thus, a CIOQ switch helps to combine the advantages of IQ and OQ switches.

IV. Impact of Input Queue Length on RED

Traditionally, AQM algorithms assume that routers have only output queues. However, as explained in the earlier section, practical switches have input queues as well, in order to lower the required speedup. This paper aims to investigate why and how an AQM algorithm will be affected by the presence of input queues in routers in addition to output queues. The question is how AQMs need to be modified to incorporate the fact that switches have both input and output buffers.

A. Effect on Congestion Measure

As mentioned before, one of the important considerations in any AQM is how exactly the level of congestion is measured at a particular router. In queue-based AQMs, the queue length is used as a measure. Consider a CIOQ switch in a congested network. The output queue will start filling as the congestion builds up. The point to note is that the effect of congestion will be felt on the input side also causing the input buffers to fill up. In other words, the effect of congestion is distributed between the input and output queues. Therefore, a reliable measure of congestion should incorporate both the queue lengths. This is particularly true when the speedup is low (around 1-1.5). If the speedup is high, then the switch will behave like an output queued switch and the input queue occupancy will be small. Thus we focus on the low speedup region.

B. Modification proposed

The modification we proposed is to use the sum of input and output lengths as the congestion measure. To be more specific, consider the RED AQM formula for the average backlog given in (1). The instantaneous queue length in this formula is replaced by the weighted sum of q_{output} and q_{input} where q_{output} is the length of the output queue and q_{input} is the sum of lengths of all VOQs corresponding to that output. In this way, the input queue length also contributes to the probability of dropping packets. The dropping itself is done only from the output queue, as the AQM is applied only to the output queue. Thus, we get:

$$Q_{avg}(i) = (1 - w_q) \times Q_{avg}(i-1) + w_q \times (\nu q_{input} + q_{output}), \quad (2)$$

where ν is the weighting factor for the input queue length. This parameter is important because it provides a means to balance the reduction in the average backlog on the one hand, and the increase in the loss rate on the other hand. The key advantage of this is that the RED (output) queue need not get filled up as much as in original RED to start reacting to congestion.

An important point to note is that, only if VOQs are used, it is easy to take into account the input queue length meant for a particular output. If instead, normal input queues are used, then it is difficult to find out how much of the input queue length is due to packets meant for a particular output. Thus, this strategy is best implemented only in VOQ based switches.

The AQM block at each output needs to know the sum of VOQ lengths corresponding to that output, besides the output queue length itself. Obtaining the sum of lengths is possible as all the queues are within the same switch.

V. Simulation Settings and Results

A. Simulation Models

To evaluate the performance of our suggested change to RED algorithm for CIOQ switches, we performed simulations in ns and compared the modified RED with the original RED algorithm as well as Droptail.

We performed simulations in a 4×4 and a 16×16 CIOQ switch. In order to simulate an $n \times n$ switch with speedup s, we set the capacities of the internal links of the switch equal to s/n times the capacity of the input link (set to 1 Mbps) since there are n internal links coming out of each input. We approximate a round-robin scheduling algorithm by allowing all the n^2 internal links to carry packets simultaneously but at a lower rate. The main difference from a real switch is that packets are sent one by one from an input in a real switch, whereas in this simulation, they can be sent together. Consequently, at an output, packets from different inputs may arrive together. The effect is that the output sees slightly more bursty traffic than in a real switch. However, this is a minor effect and can be neglected.

Each source node is capable of hosting a random number of TCP sessions. For a duration of 100 seconds, each TCP session originates from a random source and terminates at a random sink. The duration of each TCP session is randomly selected between 3 seconds and 13 seconds for the 4×4 case and between 3 seconds and 53 seconds for the 16×16. The input and output buffer sizes are both 20. Droptail is used for all the queues on the input side.

The choice of ν is crucial, and it involves a tradeoff between backlog reduction and a rise in loss rate. If it is too low, the effect of input queues will not be felt in the congestion measure. If it is too high, this would lead to too many packets getting dropped thereby reducing the utilization and increasing

the loss rate. The parameter ν is set at 0.5 for the 4×4 case and 0.125 for the 16×16 case. For the simulation, these values were chosen by trial and error. Future work could include a rigorous analysis for the optimal value of ν.

For the RED AQM, we set the min_{th} to 5 and the max_{th} to 15. The weight q_w is set to 0.004 and max_p is 0.1.

B. Simulation Results

1) Input and Output Queue Lengths: Fig. 2 displays the average input and output queue lengths when the modified RED algorithm is used in the AQM unit for the output buffer, for the 4×4 switch. The corresponding simulations were performed for original RED and Droptail, and the results are shown for comparison. For this simulation, 800 TCP sessions are started at random times. Speedup is varied from 1.1 to 2.5.

As can be seen, the input queue lengths for the three algorithms are roughly the same, and they decrease when the speedup of the switches increase. When the speedup is about 2 or higher, the average length of the input queue is very low, which implies that the switch is fast enough to handle all the input packets instantaneously, and the packets only queue at the output side.

On the output side, as expected, the Droptail queue has the longest average output queue length, since it only starts dropping when the buffer overflows, whereas RED starts dropping packets before that. Also, when the speedup is low, the output queue length for modified RED is significantly lower than that for original RED. This is because in the low speedup range, the input queues are long, as shown in Fig. 2, and when this is included in the congestion measure for the modified RED, the dropping probability is much higher than that for the original RED. However, when the speedup is high and the input queue length approaches zero, there is essentially no difference between the original RED and the modified RED, and their two output queue length curves approach each other when speedup $s \geq 2$.

Fig. 2 has demonstrated that our modification to the RED algorithm does help in reducing the average queue length for CIOQ switch, and this in turn implies that the average delay for each packet traversing through this switch is reduced.

2) Utilization and Loss Rate: We also measured the utilization and loss rate of the 4×4 switch. Speedup is fixed at 1.1 and the load is varied from 80 to 800 TCP sessions. We chose the speedup to be 1.1, because the effect of the input queue is felt only for low speedup (1 to 1.5).

Utilization is the ratio of the average number of bits received per second and the link capacity. Loss rate is the ratio of the number of packets dropped and the number of packets sent. From Fig. 3, we observe that the performance of original RED and modified RED are almost the same in terms of utilization and loss rate. This means that the gain obtained in terms of backlog in the modified RED algorithm does not cause degradation in utilization and loss rate.

We show in figures 4 and 5 the same set of results for the 16×16 switch. In this case, for the queue length simulations, 6400 TCP sessions are started at random times. Speedup is varied from 1.1 to 2.7. For the loss rate and utilization, speedup is fixed at 1.4 and the load is varied from 640 to 6400 TCP sessions. We observe again that our modified RED algorithm significantly reduces the average output queue length with marginal impact on the utilization and the loss rate.

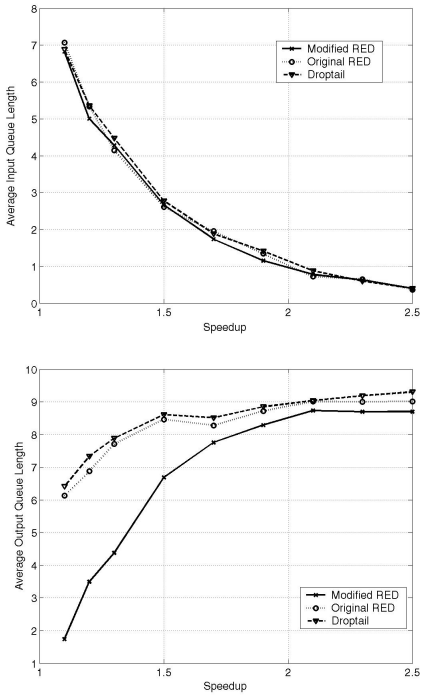

Fig. 2. Average Input and Output Queue Lengths vs Speedup for 4×4 switch (Load 800 TCP sessions)

VI. Conclusion

CIOQ switches are an important class of switches due to their good throughput and the low speedup required. In this paper, we studied the effect of input queues on the RED AQM. We have proposed the inclusion of the input queue length also in the congestion measure and have shown through simulations that this modification reduces the average backlog in the switch significantly when speedup is low. At the same time, it has been shown that the modified RED has almost the same utilization and loss rate as the original RED algorithm. Moreover, the modification that we have suggested is very simple, and it is easy to implement in practice.

Possible extension of this work includes analysis of the performance of the modified algorithm and assessment of its stability. It would also be useful to study analytically how the weighting given to the input and output queue lengths while computing the congestion measure affects the tradeoff between dropping too many packets on one extreme and not including the input queue's effects at all on the other extreme.

Acknowledgments

The authors would like to thank Prof. Dina Katabi and Dr. Supratim Deb for their valuable inputs in the course of this work.

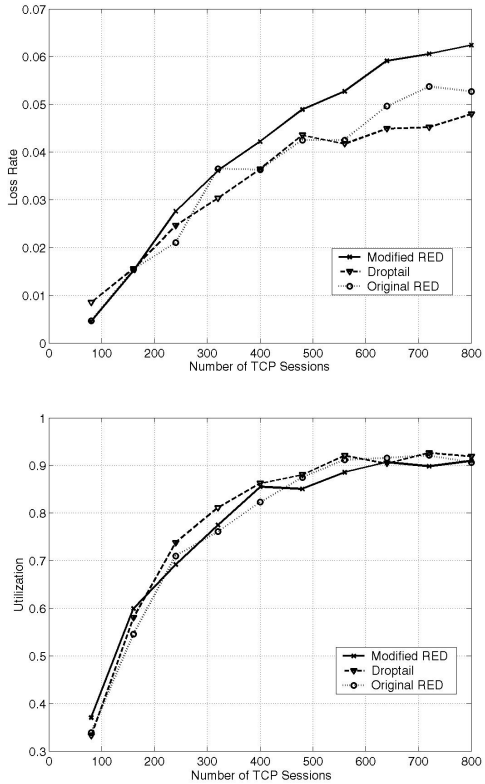

Fig. 3. Loss Rate and Utilization of the 4×4 switch at speedup $s = 1.1$.

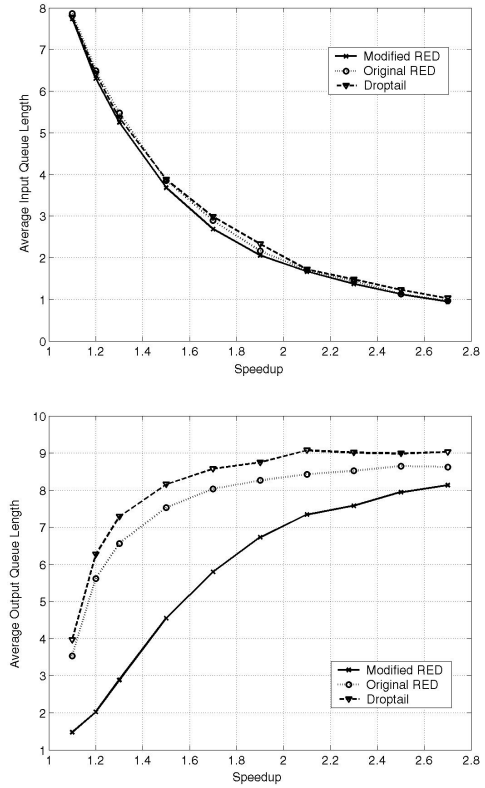

Fig. 4. Average Input and Output Queue Lengths vs Speedup for 16×16 switch (Load 6400 TCP sessions)

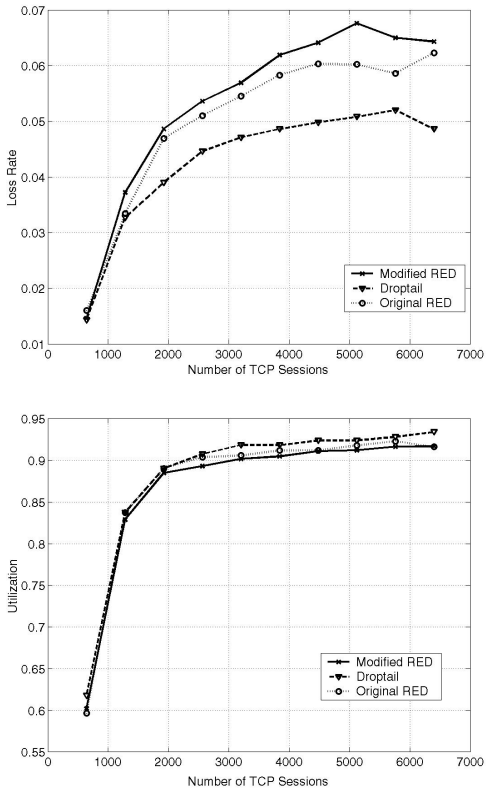

Fig. 5. Loss Rate and Utilization of the 16×16 switch at speedup $s = 1.4$.

REFERENCES

[1] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: Active Queue Management," *IEEE Network*, vol. 15, no. 3, pp 48-53, May 2001.

[2] M. Karol, M. Hluchyj and S. Morgan, "Input versus Output Queuing an a Space Division Switch," *IEEE Trans. Communications*, Vol. 35, pp. 1347-1356, 1987.

[3] B. Prabhakar and N. McKeown, "On the Speedup Required for Combined Input and Output Queued Switching," *Automatica*, Vol. 35, pp. 1909-1920, December 1999.

[4] Bartek Wydrowski and Mosche Zukerman, "Implementation of Active Queue Management in a Combined Input Output Queued Switch," *Proceedings of ICC 2003*, Vol. 1, pp. 168-172, 2003.

[5] Bartek Wydrowski and Moshe Zukerman, "GREEN: An Active Queue Management Algorithm for a Self Managed Internet," *Proceedings of ICC 2002*, New York, vol. 4, pp. 2368-2372, 2002.

[6] Wu-chang Feng, Kang G. Shin, Dilip D. Kandlur, Debanjan Saha "BLUE Active Queue Management Algorithms," *IEEE/ACM Transactions on Networking*, Volume 10, Issue 4, August 2002.

[7] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[8] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management," *Technical report, ICSI*, August 1, 2001

[9] J. Aweya, M. Ouellette, D. Y. Montuno and A. Chapman, "An Optimization-oriented View of Random Early Detection," *Computer Communications*, 2001

[10] V. Jacobson, K. Nichols, and K. Poduri, "RED in a Different Light," *Technical Report*, September 1999

[11] T. Ott, T. Lakshman, L. Wong. "SRED: Stabilized RED," *Infocom*, 1999.