AntiECN Marking: A Marking Scheme for High Bandwidth Delay Connections

Srisankar S. Kunniyur
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, Pennsylvania 19104
Email: kunniyur@ee.upenn.edu

***Abstract*— In this paper we describe a simple scheme that uses feedback from underutilized high capacity links to allow a TCP connection to aggressively increase its sending rate. The feedback is in the form of a single bit in the packet header and is given per packet. The scheme uses aggregate information to provide feedback and does not require the routers to maintain per flow state. We show through simulations that such a scheme allow TCP connections to efficiently utilize high capacity links without increasing the implementation complexity at the routers.**

I. Introduction

Throughput of a TCP connection is limited by two factors: (i) the available bandwidth in the links on its route and (ii) its own congestion window.

The first problem has been an active area of research in the community in the form of congestion control. Congestion control schemes and Active Queue Management (AQM) schemes have been proposed in recent literature [1]–[6] to study and optimize the throughput of TCP through congested links.

In this paper we consider a scenario in which a TCP connection is not limited by the bandwidth of the links on its route, but is limited by its own congestion avoidance phase. In other words, we study a scenario in which the growth of the congestion window during the congestion avoidance phase limits the throughput of the TCP connection. This problem is motivated by the presence of high capacity links (greater than 1 Gbps) in the Internet and the inability of TCP to take advantage of such high capacity links. For example, consider a TCP connection with a round-trip delay of 100ms and a segment size of 1000 bytes accessing a 1 Gbps link. To achieve a throughput close to 1Gbps, the window size of the TCP should grow to a size of 12500 segments. Assuming that slow start persists for 1000 segments, it will take around 10000 round-trip times to achieve the desired throughput. The motivation behind the congestion avoidance phase was to prevent TCP connections from increasing their rate rapidly and cause congestion (or packet drops) in the links along their route. But this motivation assumes that atleast one link in a flow's path is either highly utilized or has a small capacity. However, in the case of large bandwidth links, the slow growth of the congestion window in the congestion avoidance phase limits the throughput of the TCP connection. Thus, one would like a congestion avoidance phase that is aggressive when the available bandwidth at the links is huge and conservative otherwise. To implement such a congestion avoidance scheme, an user can either estimate the available bandwidth or rely on feedback from the routers. In the current Internet, a mechanism by which a router can signal a flow to increase its rate aggressively is absent. We provide a simple mechanism by which a router can indicate that it is under-utilized to a flow. The feedback from the router is in the form of a single bit which when set indicates that the router is underutilized and that the source can increase its rate aggressively. Since this bit achieves the opposite effect of an Explicit Congestion Notification (ECN) mark, we call it the **AntiECN (AECN)** bit or mark.

A complementary approach to improve the throughput of TCP in high capacity links has been studied in [7]. In this approach, TCP tries to estimate the available bandwidth in the links along its route by measuring the drop probability (or mark probability if ECN marking is used). Based on the measured drop probability, the response of the TCP congestion avoidance scheme is adapted. In contrast, explicit feedback from the routers is used to modify the TCP congestion avoidance phase in our approach. In [8], a new protocol called XCP is proposed which uses explicit rate information from the routers to set its rate. The scheme requires extensive changes to the present TCP protocol and router design. In this paper, we propose a scheme that uses only a single bit of feedback from the routers and require incremental changes to existing congestion-controllers and router design.

The key features of our scheme are summarized below:

1) An aggressive congestion avoidance phase when the links are underutilized. This results in significant gains in throughput
2) Performance identical to present-day TCP during periods of moderate to high congestion
3) Incremental changes to TCP and router design
4) Easily implementable at the routers

The rest of the paper is organized as follows: In Section II, we describe the algorithm for generating AECN marks as well as the implementation details at the sources and routers. We also discuss the incremental deployment of such a scheme in this section. In Section III, we present simulation results

0-7803-7802-4/03/$17.00 © 2003 IEEE

that quantify the throughput gains achieved using the AECN marking scheme. We conclude with remarks and future work in Section IV.

II. Algorithm Description

AECN marking algorithm helps TCP connections increase their congestion window aggressively by providing explicit feedback from the routers. Throughout this paper, we neglect the effect of the receiver window advertisement on TCP throughput by setting it to a very large value. The AECN marking algorithm can be divided into two parts. The first part deals with the implementation at the sources and the second part deals with the marking strategy at the router. We first start with the implementation at the sources. Though the paper is geared towards adapting TCP for high-speed links, this method in general can be used for any congestion controller [1]–[3], [7].

A. Congestion-controllers

Each packet carries a bit called the *Anti-ECN* bit in its header. The bit is initially set to zero. Each router along the packet's route checks to see if it can allow the flow to increase its sending rate. If it can, the router sets the bit to one. If the router is congested or highly utilized, it sets the bit to zero. The receiver then echoes the bit back to the sender using the ACK packet. If the bit is set to one, the sender increases its congestion window (and hence its rate) using the following equation:

$$W \leftarrow W + \frac{\Delta}{W}.$$

Note that if, $\Delta = 1$, we get the usual TCP congestion avoidance phase. If $\Delta = W$, we get the slow-start phase of the TCP applied per packet. As a result, Δ is a parameter that needs to be set at the source. In this paper, we will look at the behavior of TCP when $\Delta = W$. Note that while ECN marking is an OR operation at the link in the sense that an ECN bit is set even if one of the routers is congested, the AECN bit is an AND operation in that the AECN bit is set (at the receiver) only if all the routers in the path can support the additional increase in traffic.

B. Router implementation

When a packet with an AECN bit arrives at a router that is AECN capable, the router has to make a decision whether the flow can increase its rate in its next round-trip. A simple solution to this problem is based on the current load at the router. If the current utilization of the link is below a certain threshold called the **AECN threshold** (which can be, say a fraction ($\eta < 1$) of the capacity of the link), then the router assumes that the link is under-utilized and sets the AECN bit in the packet to one. If the current utilization is above the threshold, the router sets the AECN bit to zero. Note that this decision is made on a packet by packet basis and **the router does not need to keep any per flow state**.

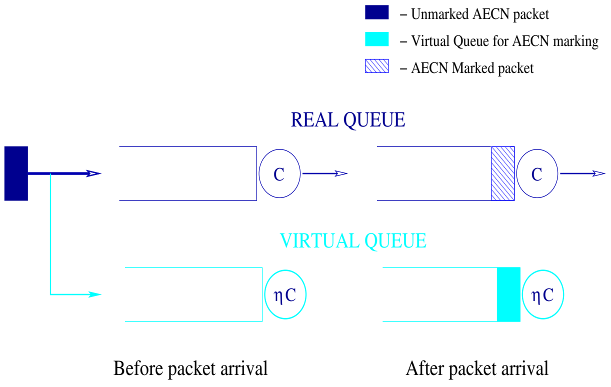

Fig. 1. AECN Marking: When the virtual queue is empty

To provide AECN marks, the router employs a scheme similar to the Adaptive Virtual Queue algorithm for ECN marking proposed in [5]. The router uses a virtual queue with a capacity equal to η times the capacity of the link and a buffer size B. On every packet arrival, the router add a fictitious packet to the virtual queue. If the virtual queue before the addition of the fictitious packet was empty, the router sets the AECN bit to one in the packet (See Figure 1). If the virtual queue was nonempty when the new packet arrived, the router sets the AECN bit to zero (See Figure 2). Define

$B =$ buffer size of the virtual queue
$s =$ arrival time of previous packet
$t =$ Current time
$b =$ number of bytes in current packet
$VQ_{AECN} =$ Number of bytes currently in the virtual queue

Then, the following pseudo-code describes the implementation at the router:

AECN Marking Algorithm

```
At each packet arrival epoch do
  $VQ_{AECN} \leftarrow \max(VQ_{AECN} - \eta * C(t - s), 0)$
      $/*$ Update Virtual Queue Size $*/$
  If $VQ_{AECN} = 0$
      Set the AECN bit to one in the packet
  else
      Set the AECN bit to zero in the packet
  endif
  $VQ_{AECN} \leftarrow VQ_{AECN} + b$
      $/*$ Update Virtual Queue Size $*/$
```

In this scheme, there are two parameters that need to be set at the router: (i) the threshold η below which the link starts providing AECN marks and (ii) the buffer size of the virtual queue. The buffer of the virtual queue has an averaging effect on the utilization. A small buffer might make the link more aggressive in marking AECN bits and hence might lead to more losses, while a huge buffer will make the link very conservative and provide no significant improvement in performance. We study the performance of the algorithm under diverse thresholds and buffer sizes using simulations.

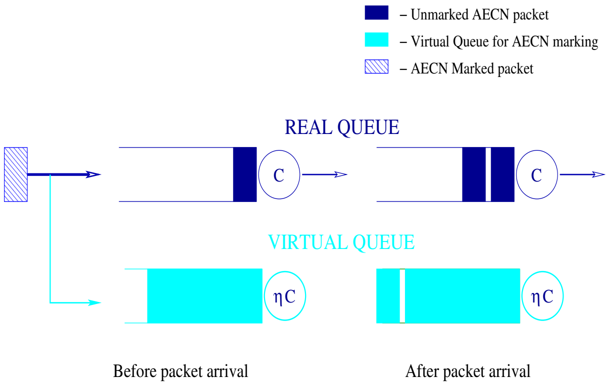

Fig. 2. AECN Marking: When the virtual queue is nonempty

C. Incremental Deployment

While AECN marking requires only incremental changes at the sources and routers, it is possible that several routers along a flow's path might not support AECN marking. At the start of a session, it is necessary for the congestion-controller to know if AECN marking is supported by all routers along its path.

To address this problem, each TCP flow incorporates a new field (say *AECN-SYN* field) in its SYN packet header which is identical to the TTL field. The values in the AECN-SYN and the TTL fields are initialized to the same value. An AECN capable router decrements this field along with the TTL field in the SYN packet. An AECN unaware router or a router that does not wish to participate in the AECN process, decrements only the TTL field. At the receiver, if the TTL field is equal to the AECN-SYN field, it is assumed that all routers are AECN capable. Presence of even a single AECN unaware or unwilling router will result in different values in the AECN-SYN and the TTL field. *Note that an router needs to look only at the SYN packet for each TCP connection.*

III. Simulations

In this section, we present simulation results that quantify the benefits of using AECN marking using the *ns-2* simulator [9]. We consider a single link network to demonstrate the benefits of AECN marking. We conduct three experiments. In the first experiment, we compare the performance of TCP with and without AECN marking for different threshold values. In the second experiment, we present results on the sensitivity of the algorithm to the virtual buffer size. In the third experiment we consider three TCP sources accessing a common link and present results on the fairness and utilization of the link.

Experiment 1

In this experiment, we consider a single link of capacity 1 Gbps. We then study the throughput characteristics of a single TCP user accessing the link for different threshold values. Due to space constraints, we present the results for only two different values of round-trip delay: 40ms and 200ms. We assume that the TCP user is ECN enabled and that the link provides AECN marking whenever the throughput of the link falls below a certain threshold called the **AECN threshold**. In this experiment, we fix the size of the virtual queue to be equal to the bandwidth delay product. We use the AVQ algorithm proposed in [4], [5] (with a desired utilization of 0.96) as the AQM scheme that provides ECN marks. Note that one can also use RED [10], REM [11], PI [12] or any other AQM scheme to provide ECN marks.

We plot the evolution of the congestion window of a TCP connection with and without AECN marking in Figure 3. The round-trip delay is set to 40ms and the virtual queue size is set to 5000 packets. The AECN threshold (η) is varied from 0.40 to 0.75 in this simulation. From Figure 3 we can see that an AECN user is able to utilize the link more efficiently than an user without AECN marking. The amount of time that the congestion window is increased aggressively depends upon the AECN threshold which is used to set the AECN marks at the router. A higher threshold results in more AECN marks which results in an aggressive congestion window increase for a longer period of time. A higher threshold value, while leading to a greater throughput increase might also lead to instabilities in the network in the presence of a large number of flows. Design of the appropriate threshold value is a topic for future research. Note that in the experiment there are no losses for any threshold setting at the link. The congestion window is cut in half when the link marks a packet (ECN). Also, note that the rate at which the congestion window is increased just before a packet is marked (ECN) is identical to the normal TCP operation. The utilization at the link is shown in Figure 4. We can see that by using AECN marking, one can improve the throughput of the TCP connection considerably. Also, note that when the utilization at the link is greater than the threshold, no AECN marks are generated. As a result, the congestion window is increased by one packet per round-trip time. In other words, when the utilization at the link is above the AECN threshold, the behavior of an AECN user is identical to an user without AECN. The congestion window evolution and the utilization at the link when the round-trip delay and the virtual queue size is increased to 200 ms and 25000 packets respectively are shown in Figures 5 and 6. Similar conclusions hold in this case.

The percentage improvement in overall throughput when AECN marking is employed is shown in Table I. We can see that performance improvement is significant when the bandwidth delay product is high. A 30% improvement in throughput can be obtained using AECN marking.

Experiment 2

In this experiment we look at the sensitivity of the AECN scheme to the size of the virtual buffer used for AECN marking at the link. A small size of the virtual buffer would make the link aggressively use the AECN marking to improve its utilization while a larger value would make the link more conservative in its AECN marking. Also, a smaller value would make the link sensitive to bursts. In this experiment, we presents results that show the sensitivity of the AECN

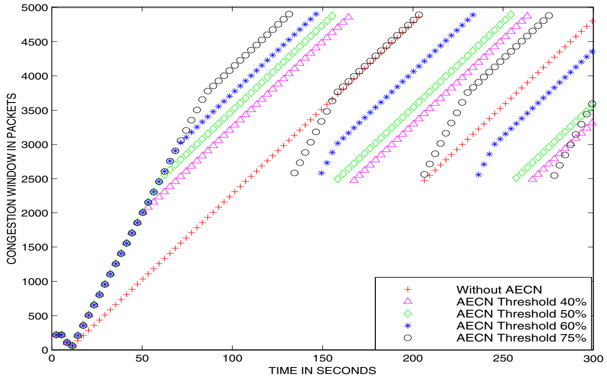

Fig. 3. Evolution of the congestion window with and without AECN marking for a RTT of 40ms

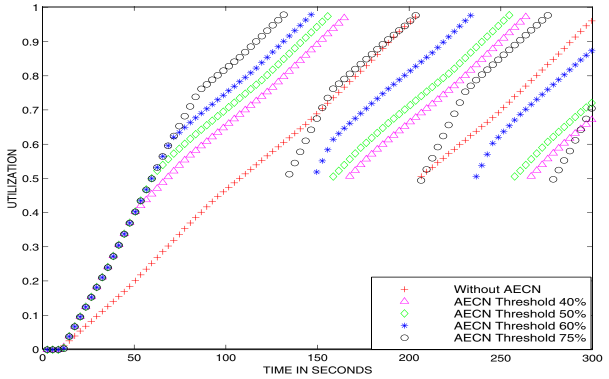

Fig. 4. Evolution of the utilization at the link with and without AECN marking for a RTT of 40 ms

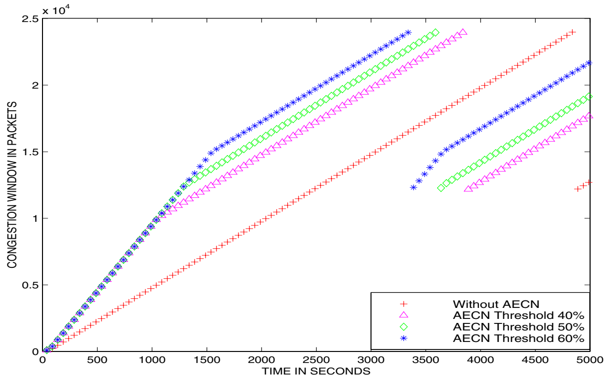

Fig. 5. Evolution of the congestion window with and without AECN marking for a RTT of 200 ms

	Percentage Improvement	
	40 ms	200 ms
No AECN	–	–
AECN Threshold 40%	10%	17%
AECN Threshold 50%	12%	20%
AECN Threshold 60%	17%	26%
AECN Threshold 75%	19%	29%

TABLE I

PERFORMANCE IMPROVEMENT USING AECN MARKING

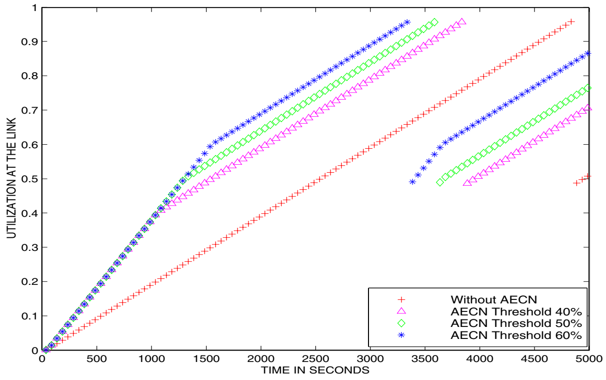

Fig. 6. Evolution of the utilization at the link with and without AECN marking for a RTT of 200ms

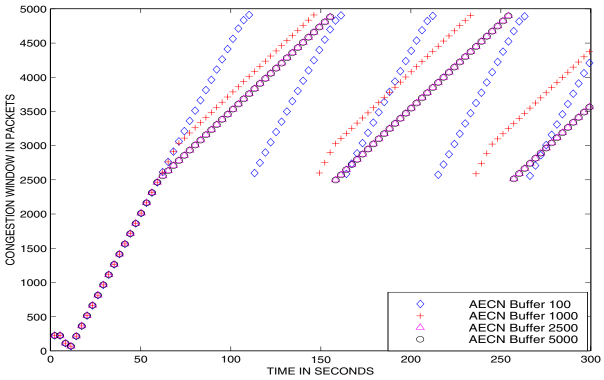

Fig. 7. Evolution of the congestion window with AECN marking for different values of virtual buffer size

algorithm to the virtual buffer size. The evolution of the congestion window for different buffer sizes are shown in Figure 7. The round-trip delay is set to 40ms and the AECN threshold is set at 50%. We start with a virtual buffer size of 5000 packets which is equal to the bandwidth delay product of the flow. We then reduce the buffer size to 2500, 1000 and 100 packets. When the virtual buffer size decreases, sources increase their congestion window aggressively for a longer period of time. This is due to the fact that the link reacts quickly to changes in utilization. As a result, smaller virtual buffers lead to higher link utilizations. However, performance of the AECN marking algorithm with very small virtual buffer sizes in the presence of short flows is a topic of future research.

Experiment 3

In this experiment we consider three TCP flows that access a high bandwidth link of capacity 1 Gbps. We assume that all the three connections have an identical round trip delay of 40ms. We also assume that either all connections use AECN marking (with a threshold of 50% and virtual buffer size equal to the bandwidth delay product) or none of them employ AECN marking. The starting time of each flow is uniformly distributed between 0 and 10s. Due to space constraints, we

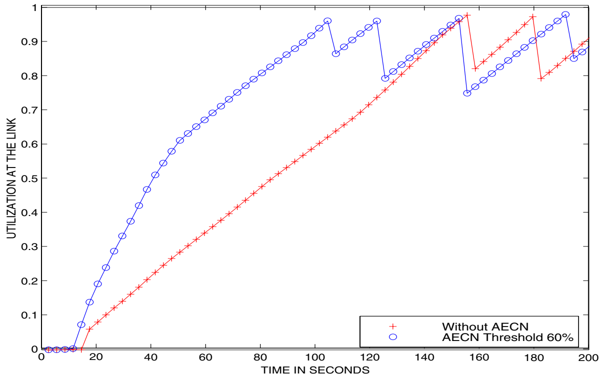

Fig. 8. Total utilization at the link with and without AECN marking

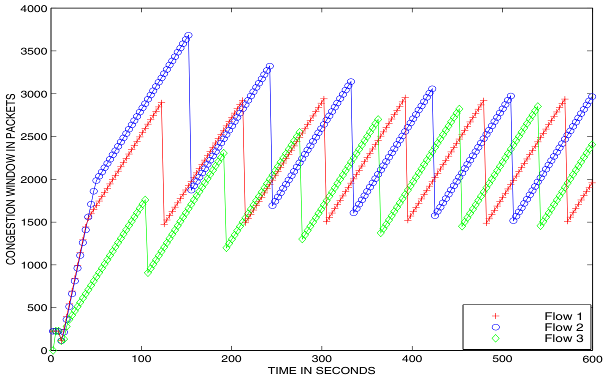

Fig. 9. Evolution of the congestion window when all the three TCP connections use AECN marking

present only one among several sets of experiments that we conducted to validate our results. In Figure 8, we compare the utilization at the link when all the connections employ AECN marking with the utilization at the link when none of the connections employ AECN marking. We can see that the link is utilized more efficiently when AECN marking is employed. The evolution of the congestion windows of all the three flows is shown in Figure 9. Due to the random starting times of the flows, there is an initial discrepancy in the congestion window sizes of the flows. However, after an initial transient, the congestion windows of the flows become roughly identical. This shows that the AECN marking preserves the fairness of the TCP algorithm when all the flows have identical round-trip delays. Fairness properties of the AECN scheme when users with different round-trip delays access the same link is a topic for future research. The congestion window evolution when none of the TCP connections employ AECN marking is shown in Figure 10 for comparison.

IV. Conclusions

In this paper we presented a simple marking scheme called the **AntiECN (AECN) marking** that allows TCP connections to aggressively increase their rate using minimal feedback from the routers. The feedback is in the form of a single

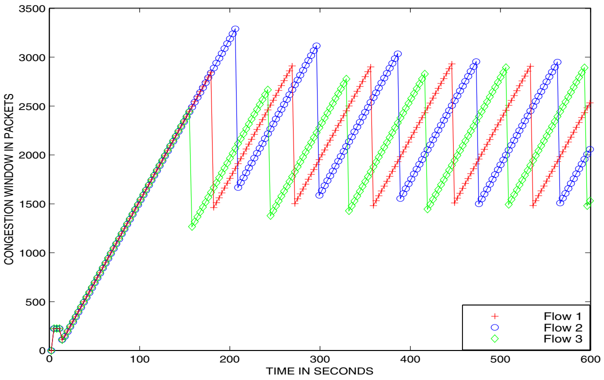

Fig. 10. Evolution of the congestion window when all the three TCP connections are not AECN capable

bit and is done on a *per packet* basis. The implementation complexity at the routers as well as the sources is small and the scheme does not require the routers to maintain per-flow state. In the presence of a congested link, the behavior of a TCP connection with AECN marking is identical to a TCP connection without AECN marking.

The simulation results show that one can achieve significant throughput gains when using AECN marking. The implementation of the AECN algorithm requires the choice of the increase parameter at the source and the threshold and the virtual buffer size at the link. Optimal choices of these parameters and their effect on throughput and fairness is a topic for future research.

References

[1] R. Gibbens and F. Kelly, “Resource pricing and the evolution of congestion control,” 1998, preprint.

[2] S. H. Low and D. E. Lapsley, “Optimization flow control I: Basic algorithm and convergence,” *IEEE/ACM Transactions on Networking*, pp. 861–875, December 1999.

[3] S. Kunniyur and R. Srikant, “End-to-end congestion control: utility functions, random losses and ECN marks,” in *Proceedings of INFOCOM 2000*, Tel Aviv, Israel, March 2000, Also to appear in IEEE/ACM Transactions on Networking, 2003.

[4] ——, “A time-scale decomposition approach to adaptive ECN marking,” *IEEE Transactions on Automatic Control*, vol. 47, pp. 882–894, June 2002.

[5] ——, “Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management,” in *Proceedings of SIGCOMM 2001*, San Diego, CA, August 2001.

[6] C. Hollot, V. Misra, D. Towlsey, and W. Gong, “On designing improved controllers for AQM routers supporting TCP flows,” in *Proceedings of INFOCOM 2001*, Anchorage, Alaska, April 2001.

[7] S. Floyd, “Highspeed TCP for large congestion windows,” internet draft draft-floyd-tcp-highspeed-00.txt, Preprint, June 2002.

[8] D. Katabi, M. Handley, and C. Rohrs, “Internet congestion control for future high bandwidth-delay product environments,” in *Proceedings of ACM Sigcomm*, 2002.

[9] ns 2 (online), http://www.isi.edu/nsnam/ns.

[10] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, August 1993.

[11] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, “REM: Active queue management,” *IEEE Network*, pp. 48–53, June 2001.

[12] C. Hollot, V. Misra, D. Towlsey, and W. Gong, “A control theoretic analysis of RED,” uMass CMPSCI Technical Report 00-41, 2000.