Achieving High Throughput in Low Multiplexed, High Bandwidth, High Delay Environments[*]

Abhinav Kamra
Dept. of Computer Science
Columbia University
New York, NY 10027
kamra@cs.columbia.edu

Vishal Misra
Dept. of Computer Science
Columbia University
New York, NY 10027
misra@cs.columbia.edu

Don Towsley
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
towsley@cs.umass.edu

Abstract

In this paper we explore combining ECN bits and loss feedback to achieve a high throughput connection in a high bandwidth, low multiplexed environment. In our scheme, a TCP source reacts differently on receiving an ECN than it does on infering a packet loss in terms of congestion window update. Coupled with an AQM scheme such as RED [FJ93], TCP sources achieve high utilization at a bottleneck router, but at the cost of fairness. We introduce a lightweight AQM mechanism that ensures fairness while still providing high utilization to the sources. Our proposal is complementary to the HSTCP [FRS02] modification suggested by Floyd et al. Simulations results indicate that we can achieve near 100% utilization even with a single source over gigabit links.

1 Introduction

As noted by other researchers [FRS02, KHR02, Kel02], the congestion control mechanism of TCP limits the performance in high bandwidth environments. The causes are many, for instance the loss-throughput formula of TCP puts unrealistic demands on the loss rates needed to achieve high throughputs of; say, a gigabit or above. In realistic environments, such low loss rates (of the order of 10^{-8} for 1Gbps Throughput) are not possible, and other layers of the protocol stack can provide loss rates significantly higher than that. Related to this is the fact that, when a loss does occur in a low multiplexed, high bandwidth environment, considerable time is required for a flow to reclaim the lost bandwidth. The bandwidth is lost due to TCP's congestion control mechanism, due either to a timeout event or a window reduction event. In any case, an unreasonably long time is required for TCP to get back to sending at available capacity.

Floyd et. al [FRS02] proposed HSTCP which has a more aggressive alternate linear response function for TCP in high bandwidth environments. We present a proposal that is complimentary to HSTCP. We work under the assumption that the path in question has ECN capable routers. We then take the point of view that an ECN bit reflects "mild" congestion, and should be treated differently

[*]This work was supported in part by the Department of Energy under grant DE-FG02-02ER25528

from a loss event. Building on this, we propose simple modifications to base TCP algorithms that both increase performance as well as retain fairness in the protocol. We call our protocol TCP/E, for ECN enhanced or ECN aware. Preliminary simulations indicate that our modifications yield a high utilization, fair protocol that scales well with increasing bandwidth as well as delays.

2 Protocol Description

We start with the notion that the presence of an ECN mark is indicative of less severe congestion than the presence of a packet loss. An ECN bit indicates that there is still space in the bottleneck buffer to accommodate the particular packet, whereas a drop due to buffer overflow indicates extreme congestion. Once, we make this conceptual difference, we treat an ECN bit differently than a loss indication. Note that this is in variance with the original spirit of the ECN proposal, which mandates that an ECN bit be treated the same as a triple duplicate ack and hence result in a decrease of congestion window by a multiplicative factor, usually $1/2$. In our proposal, the reaction of a TCP source to an ECN bit in the ack is to *freeze* the congestion window at the current value, rather than reduce it multiplicatively.

With a correctly designed AQM in place, it is easy to imagine what happens with a single source. The congestion window of the flow starts growing, and gradually it starts receiving ECN marks which slows the growth. Eventually, the ECN marking probability reaches 1, at which point, "rate matching" has occurred between the sender and the link, and the sender transmits at *exactly* the link speed. To achieve this behavior, a modification is also needed at the TCP receiver. Normally, a TCP receiver echoes the ECN bit in the ack packet until it has received an acknowledgment from the sender that the congestion window has been reduced. Since in our proposal we do not reduce the congestion window in response to an ECN bit, this is not necessary. Moreover, for our scheme to function properly, there has to be a one to one correspondence between the ECN marks in the forward direction and the ECN marks reflected in the ack packets. Hence, a receiver simply reflects the ECN bit in a *single* ack packet on receipt of an ECN marked data packet.

While this modification achieves near 100% utilization with a single source over high bandwidth, high delay links, it can be grossly unfair if more than one flow is sharing the link. For example, consider the scenario where two flows share the link, with one flow starting up much later than the other one. By the time the second flow starts up, the first flow has already achieved the "rate matching"; hence the ECN marking probability at the link AQM is (close to) 1. Nearly all the acks that the second flow now receives have the ECN bit set and its congestion window cannot grow. This results in the first flow obtaining most of the bandwidth. Thus the modification introduces a "temporal unfairness", flows that start out earlier get a larger share of the bandwidth.

To address the issue of fairness, we randomly introduce losses. If we simply introduce Bernoulli losses as part of the AQM (e.g., with a small probability an ECN mark is replaced by a packet drop), then flows with a higher share of the bandwidth will receive more losses. The losses would induce multiplicative reductions in the sending rate; hence the bandwidth would be distributed fairly over time. To ensure that the loss feedback is sent more aggressively to flows that grab an unfair share, we implement a CHOKE [PPP00] like mechanism. At periodic intervals, an incoming packet is matched with a random packet in the buffer, a match is successful if the two packets belong to the same flow. The packet is dropped on a successful match.

Bernoulli losses result in a multiplicative reduction in the sending rate of a flow. For a low level of multiplexing, the time required to reclaim the lost bandwidth grows as the link bandwidth and propagation delay grows. Hence, fairness comes at the expense of loss in utilization. For high bandwidth links, this reduction in utilization can be quite severe. In the final version of TCP/E we again exploit the ability to generate ECN marks. We treat the presence of an ECN mark again as mild congestion or onset of congestion. If a flow receives an ECN mark in the ack, then besides freezing the congestion window, it also switches to congestion avoidance. If a flow receives no ECN mark over a round trip time (or a time out interval, in any case a time constant that is based upon the measured round trip delay), then the flow switches to slow start until it receives an ECN mark again (or a packet gets dropped). Thus, a flow is able to quickly reclaim lost bandwidth due to the multiplicative reductions. For this scheme to work, the AQM in place should be *fast* and reactive, i.e. it should not keep a lot of history. One of the fastest AQM schemes is RED with no averaging, and in our experiments in the next section we primarily use RED based on an instantaneous queue length rather than an averaged one. Thus, our feedback mechanism combines CHOKE (for the drops) and RED with Instantaneous queue (for the ECN marks). Subsequently we refer to this policy as CHOKE/RED-I.

```
OnAck (ack)
{
      // Slow-Start or Congestion Avoidance
      increase = (mode == SS)?(1):(1/cwnd);
      if(ECNset(ack)) // ECN marked ack
      {
               lastECNtime = Currenttime;
               // Switch to Congestion Avoidance
               mode = CA;
      }
      else
      {
               cwnd += increase;
               if(Currenttime - lastECNtime > RTT)
               // Switch to Slow-Start
                    mode = SS;
      }
      DefaultAction(ack);
}
```

3 Simulations

We now present some simulation results to evaluate the performance of TCP/E. All the simulations were performed using the ns-2 [ns] Network Simulator. Unless otherwise stated, all simulations are with an OC-48(2.4 Gbps) bottleneck link, a bottleneck buffer of 32 Megabits (so a peak delay of 50 ms can be added by the buffer), and 60ms of round trip propagation delay. Data packets are all 1000 Bytes.

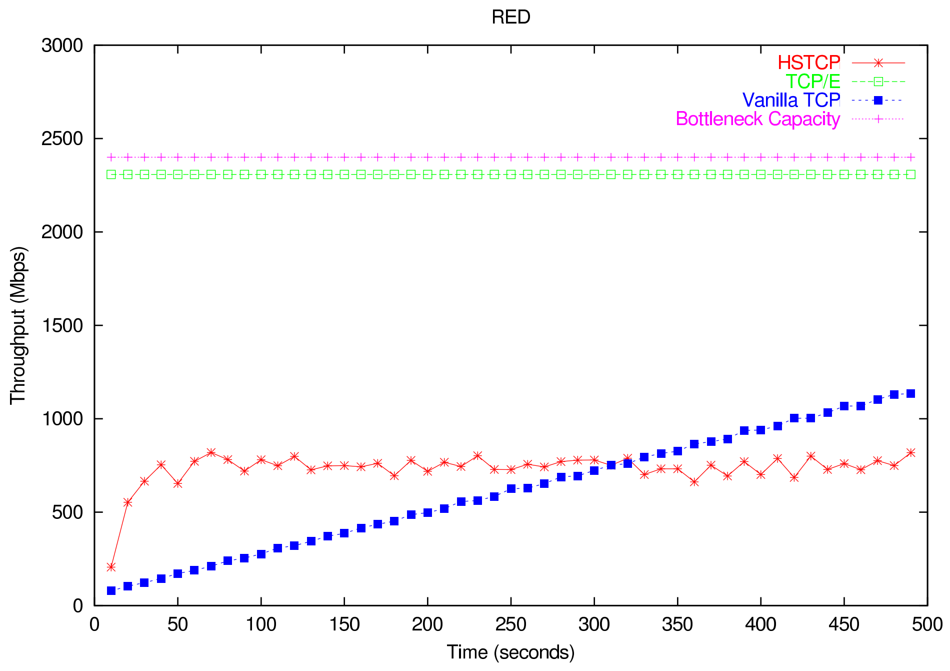

Figure 1: TCP Throughput for TCP variants with RED-I AQM policy

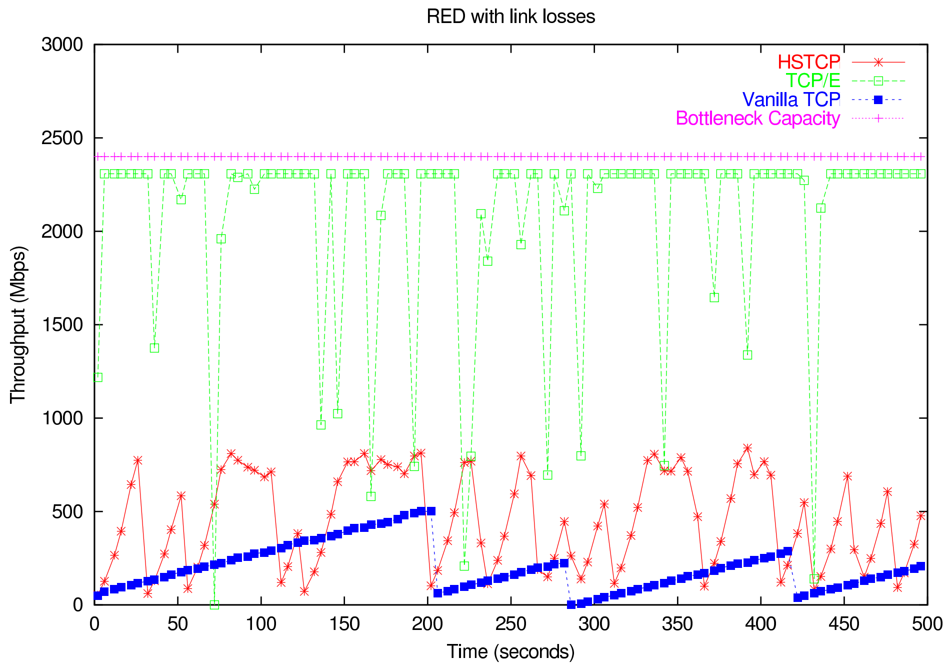

Figure 2: TCP Throughput for TCP variants with RED-I AQM policy and random link losses

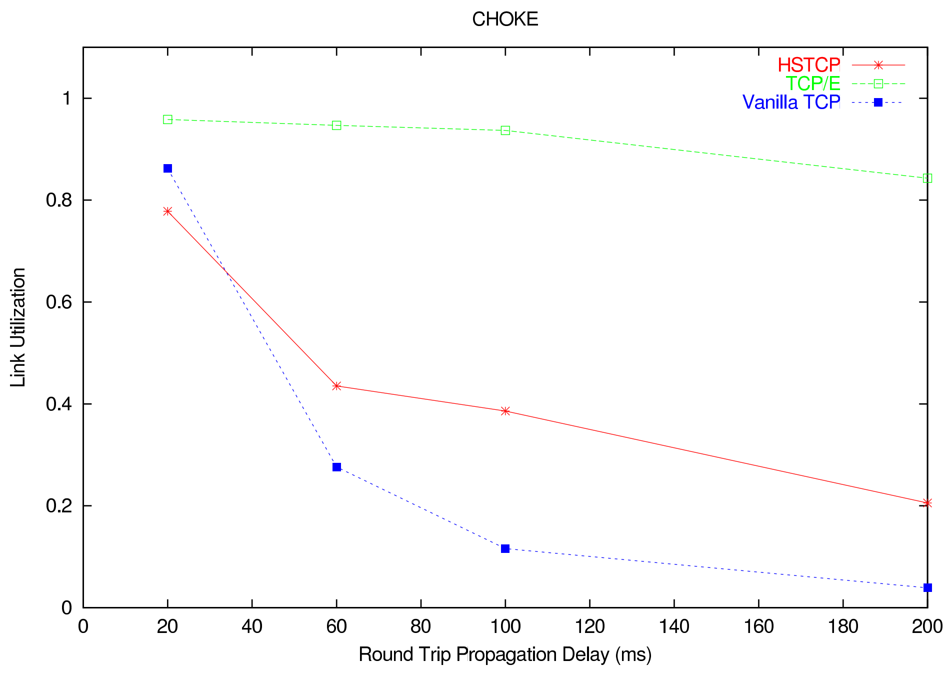

Figure 3: Link Utilization vs Delay for the TCP variants with CHOKE/RED-I AQM policy

3.1 Link Utilization

Figure 1 shows a sample TCP performance for the different TCP variants with RED as the AQM policy. In each (separate) experiment, a single TCP variant is transmitting over the OC-48 link. We observe that TCP/E is able to transmit at a sustained rate equal to the link capacity. Actually its slightly less than the link capacity because we measure the rate of transfer of only the TCP header+payload rather than the whole packet. Figure 2 shows the performance when the links are lossy. The loss probability was configured to provide one loss approximately every 10 seconds.

Figures 3, 4, 5, 6 show the average link utilization of the TCP variants with different AQM scheme variants and varying round trip propagation delays. In each experiment three flows of the same TCP variant compete for bandwidth share on an OC-48 link. In the absence of random losses on the link, TCP/E achieves near 100% utilization regardless of the propagation and queuing delays. As we introduce random losses on the link, TCP/E still achieves a significantly higher utilization than the other TCP variants under various AQM schemes.

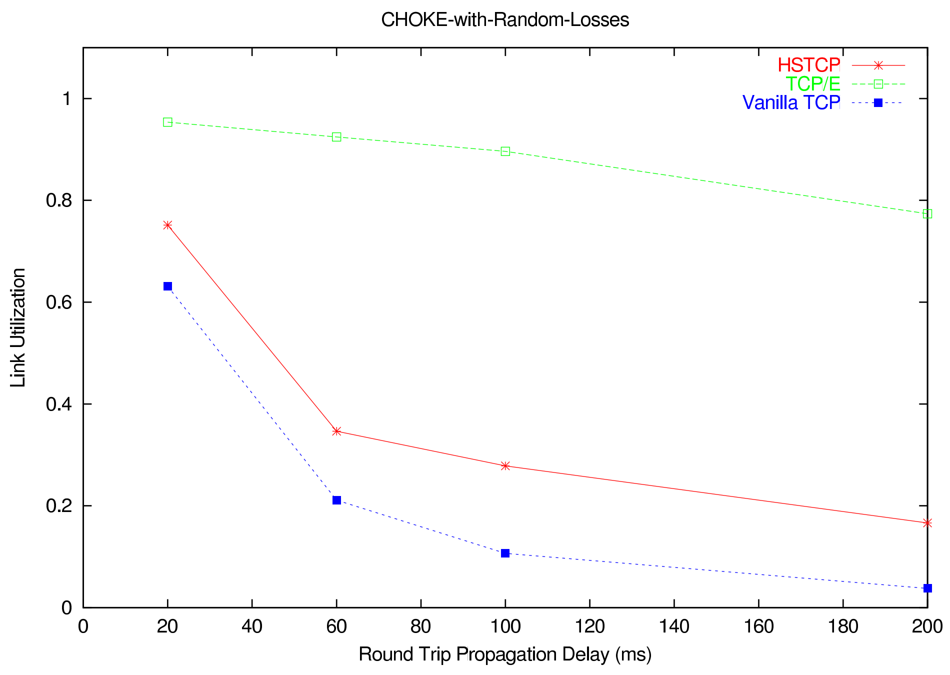

Figure 4: Link Utilization vs Delay for the TCP variants with CHOKE/RED-I AQM policy and random losses on the link

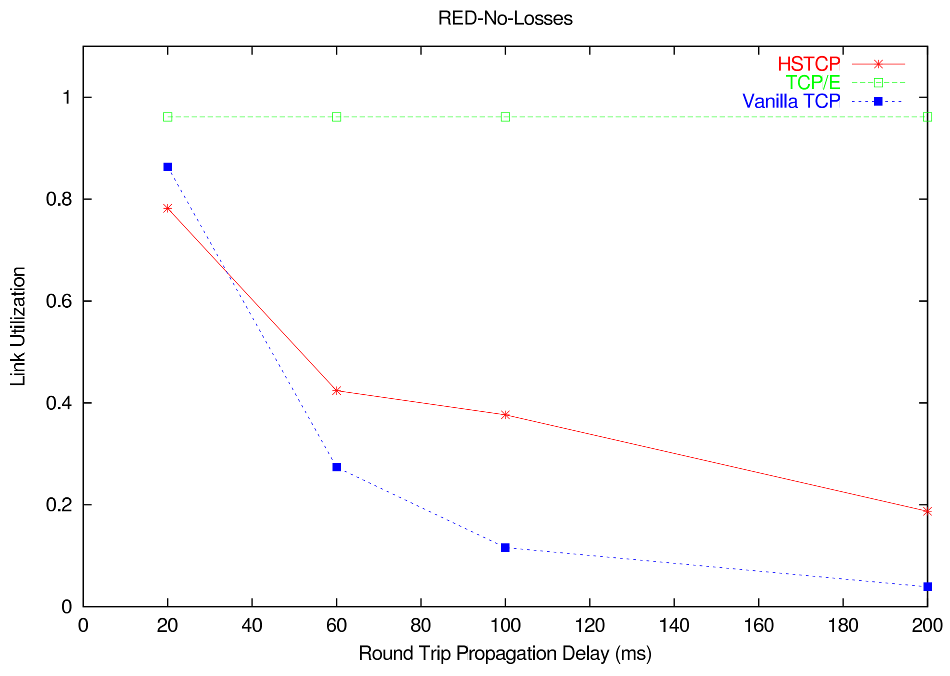

Figure 5: Link Utilization vs Delay for the TCP variants with RED AQM policy

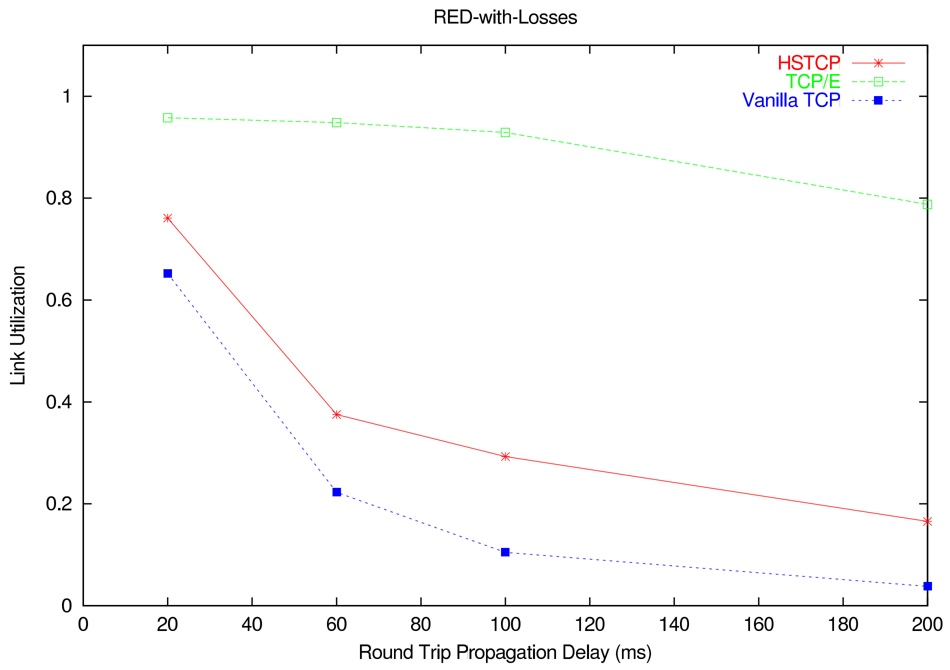

Figure 6: Link Utilization vs Delay for the TCP variants with RED AQM policy and random losses on the link

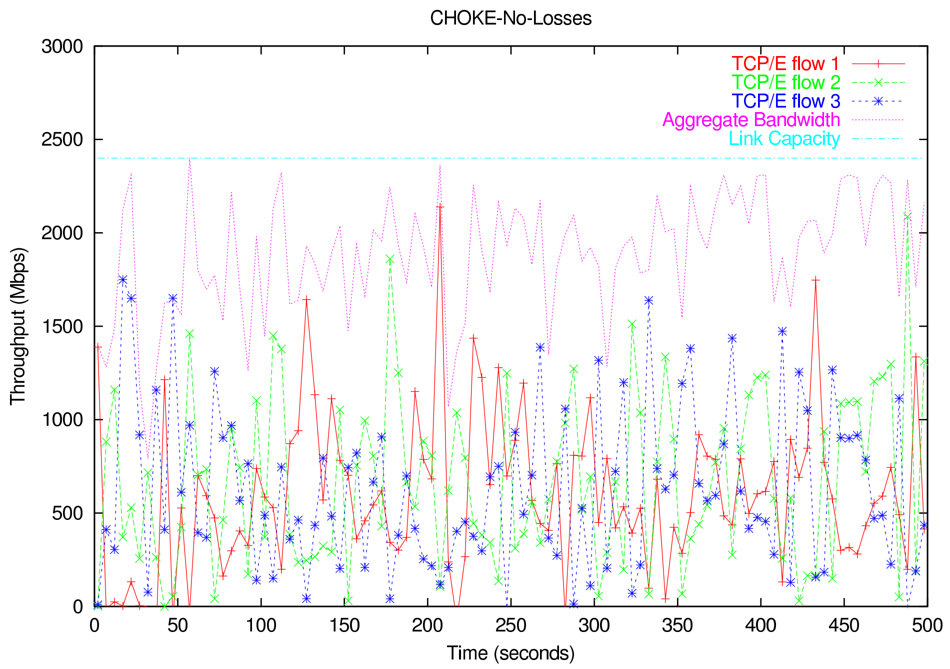

Figure 7: TCP Throughput for 3 TCP/E flows competing for bandwidth share. The AQM policy is CHOKE/RED-I

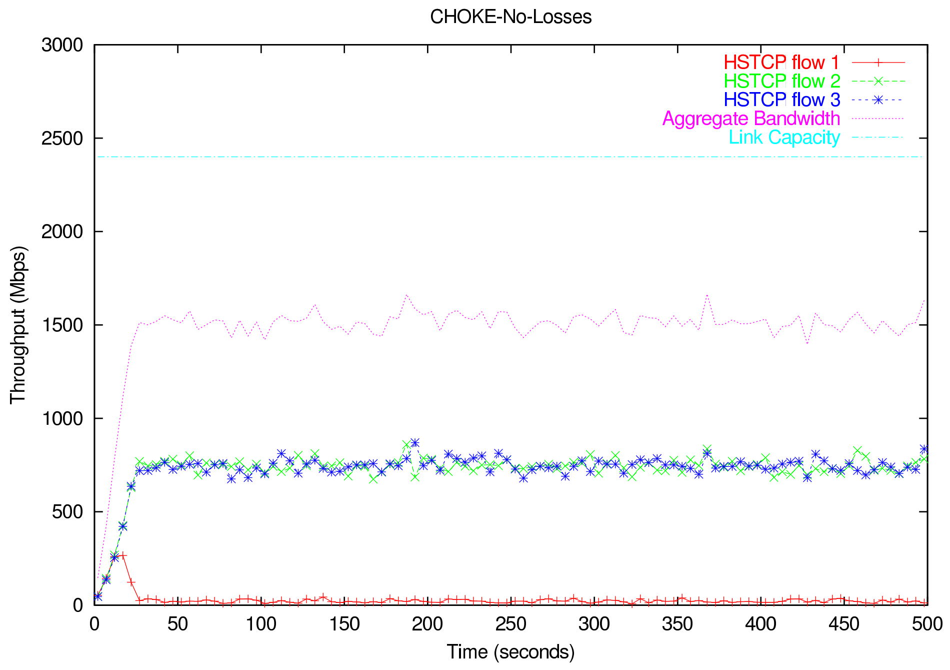

Figure 8: TCP Throughput for 3 HSTCP flows competing for bandwidth share. The AQM policy is CHOKE/RED-I

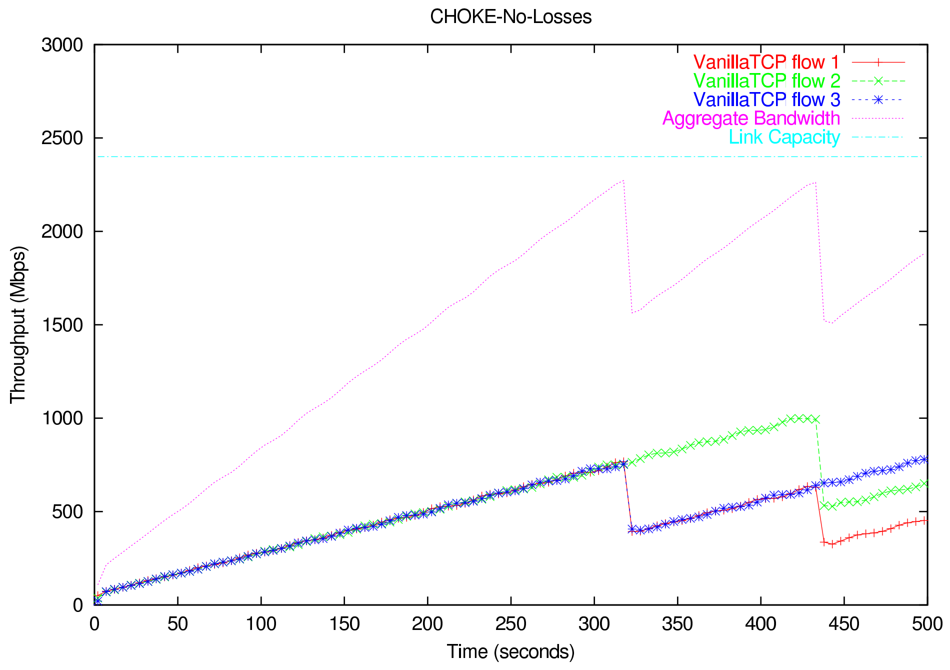

Figure 9: TCP Throughput for 3 Standard TCP flows competing for bandwidth share. The AQM policy is CHOKE/RED-I

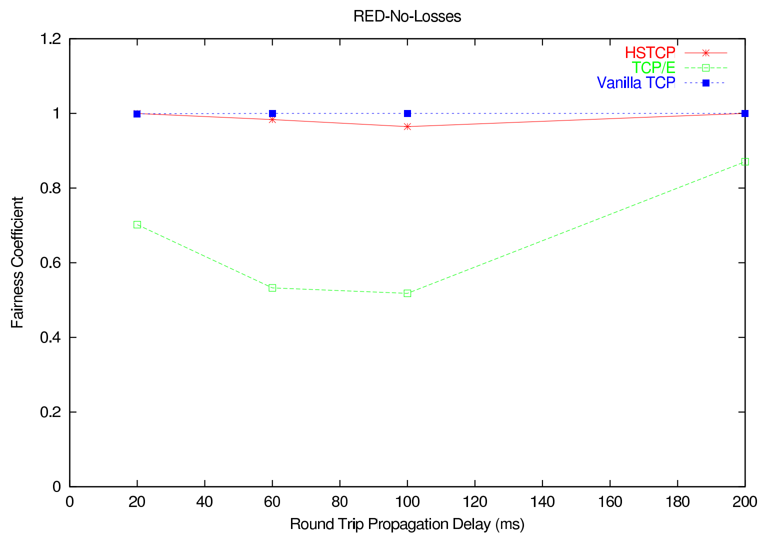

Figure 10: Fairness Coefficient vs Delay for the TCP variants with RED AQM policy

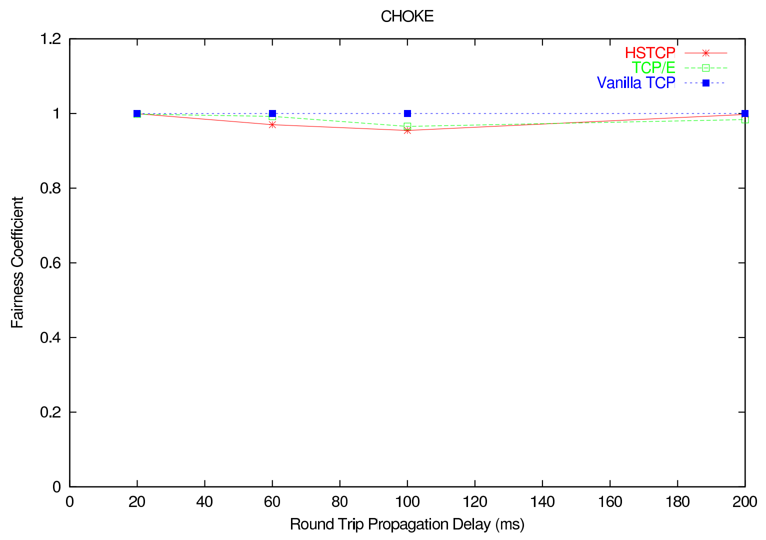

Figure 11: Fairness Coefficient vs Delay for the TCP variants with CHOKE/RED-I AQM policy

3.2 Fairness

Figure 7 illustrates the throughputs of three competing TCP/E flows smoothed over a 500 second simulation time. Figures 8, 9 show similar results for HSTCP and standard TCP respectively.

In Figures 10, 11, 12, 13 we plot the Fairness Coefficient [JCH84] achieved by the TCP variants for varying round trip propagation delays and AQM policies. From Figure 10 we observe that, unless we introduce Bernoulli losses or a CHOKE like mechanism, TCP/E can be significantly unfair in bandwidth sharing. Once that is in place, TCP/E is as fair as any other TCP variant. Note that TCP/E will be *unfair* to other TCP variants because it uses the bandwidth available more efficiently, and we do not claim that it is "TCP-fair" in that sense.

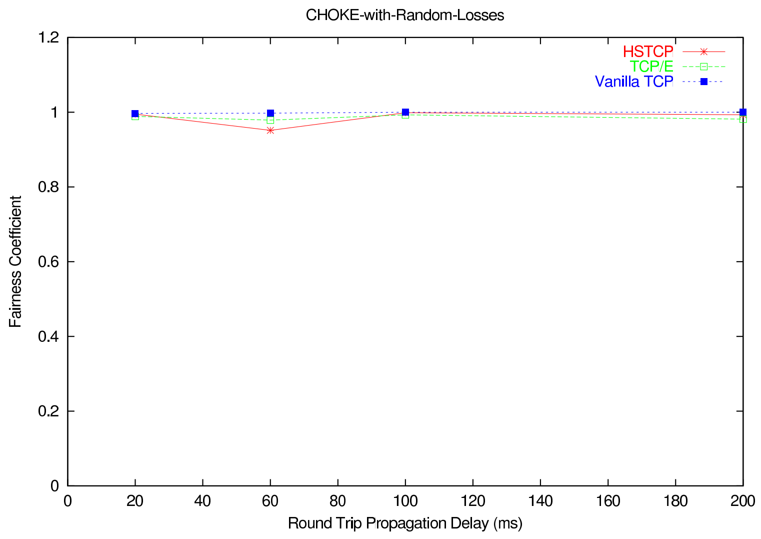

Figure 12: Fairness Coefficient vs Delay for the TCP variants with CHOKE/RED-I AQM policy and random losses on the link

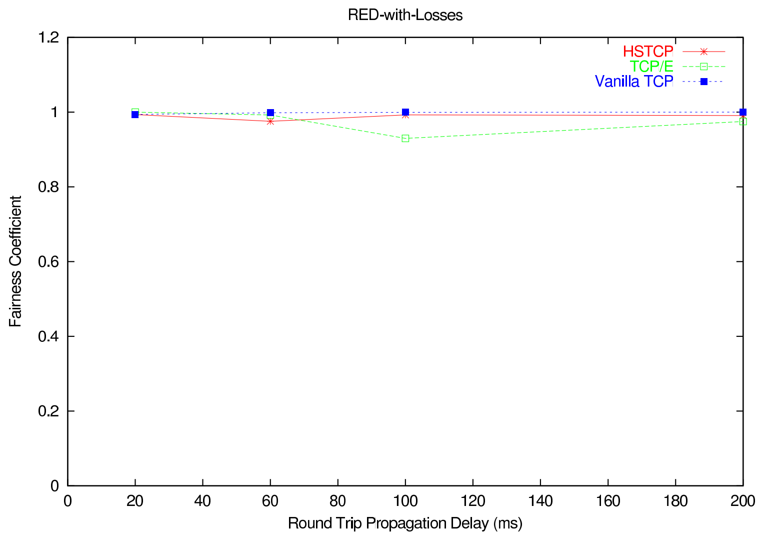

Figure 13: Fairness Coefficient vs Delay for the TCP variants with RED AQM policy and random losses on the link

4 Conclusions

We have proposed modifications in TCP's response to ECN marked acknowledgements in order to improve its performance in high bandwidth, high delay networks. We further suggest, as an AQM scheme, a combination of CHOKE and RED to provide fair bandwidth allocation to competing TCP/E flows.

Simulations were used to analyze the new design and to compare with existing protocols. The results indicate that TCP/E provides high link utilization in high bandwidth environments even with lossy links. With the new CHOKE/RED-I AQM scheme in place, competing TCP/E flows obtain a stochastically fair share of the bandwidth.

References

[FJ93] Sally Floyd and Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1993.

[FRS02] Sally Floyd, Sylvia Ratnasamy, and Scott Shenker. Modifying TCP's Congestion Control for High Speeds. http://www.icir.org/floyd/hstcp.html, 2002.

[JCH84] Raj Jain, D. Chiu, and W. Hawe. A Quantitative Measure of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. DEC Research Report TR-301, September 1984.

[Kel02] Tom Kelly. A scalable TCP with adaptive congestion detection at routers. In *Proceedings of IEEE Computer Communications Workshop*, 2002.

[KHR02] D. Katabi, M. Handley, and C. Rohrs. Internet Congestion Control for Future High Bandwidth-Delay Product Environments. In *Proceedings of ACM/SIGCOMM*, 2002.

[ns] ns-2 Network Simulator. http://www.isi.edu/nsnam/ns/.

[PPP00] R. Pan, B. Prabhakar, and K. Psounis. CHOKe, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. In *Proceedings of IEEE Infocom*, 2000.