Performance Sensitivity and Fairness of ECN-Aware ‘Modified TCP’

Archan Misra[1] and Teunis J Ott[2]

[1] IBM T J Watson Research Center,
19 Skyline Drive, Hawthorne, NY 10532, USA.
archan@us.ibm.com
[2] Department of Computer Science,
New Jersey Institute of Technology, Newark, NJ 07102, USA.
ott@oak.njit.edu

Abstract. The paper discusses how Explicit Congestion Notification (ECN) can be used to devise a congestion control mechanism for the Internet, which is more rapidly reactive and allows best-effort flows to rapidly adjust to fluctuations in available capacity. Our ECN-mod protocol involves simple modifications to TCP behavior and leverages more aggressive marking-based router feedback. Simulations show that ECN-mod is better than TCP NewReno even for Web-style intermittent traffic sources, and makes the link utilization significantly less sensitive to the variation in the number of active flows. Simulations also show that, while ECN-mod flows obtain a larger portion of the available capacity than conventional best-effort traffic, they do not starve or significantly penalize such TCP-based flows.

1 Introduction

The advantages of using Explicit Congestion Notification (ECN) [1, 2] to provide unambiguous congestion feedback to adaptive (TCP) Internet traffic are well documented in literature. We, however, believe that the full benefit of ECN-capable routers has not been effectively realized: *a much more powerful and responsive congestion control framework can be developed if TCP is modified to differentiate between packet marking and packet losses.* When faced with rapid variations in the available bandwidth, such a modified *rapidly reactive* protocol must possess two conflicting characteristics:

- During congestion, the adaptive flows must backoff rapidly to prevent *congestion collapse.*
- Whenever additional bandwidth becomes available, flows should rapidly increase their transmission rate to avoid under-utilization of available capacity.

The design of such an ECN-aware TCP-like protocol for rapid adaptation to variable capacity was presented in [3, 4], which suggested that modifications to TCP behavior must be designed in tandem with marking behavior in routers. The protocol modifications exploit the fact that packet *marking* probabilities can be made as high as 100% without causing any undesirable behavior (as opposed to packet *dropping* probabilities which need to be restricted to $\sim 10 - 15\%$.)

In this paper, we first provide a brief recap of our suggested ‘ECN-mod’ window adaptation algorithm, placing it in the context of a generalized class of 'polynomial' [5] window adaptation algorithms. In particular, we investigate the tradeoffs resulting from a choice of the various coefficients of the polynomial window adjustment procedure. We then report on the result of extensive simulation studies that investigate the properties of our protocol.

Unlike our earlier studies in [4], which used persistent TCP sources, we first use bursty "Web-like" TCP sources and observe the performance of our 'ECN-mod' protocol. As with our earlier observations with persistent TCP sources, we see that in contrast to current ECN-aware TCP NewReno, ECN-mod flows can achieve better link utilization when faced with rapidly changing available bandwidth. However, the improvement in utilization is not as dramatic as with persistent TCP traffic. Using a further set of simulation studies, we show a far more important benefit of the use of the 'ECN-mod' protocol– it makes the network performance much less sensitive to variations in the actual traffic loads. ECN-mod flows (unlike the current ECN-aware TCP flows) are able to operate well even when the marking rates are as high as $\sim 80-90\%$; previous research [6, 7] has clearly documented why such aggressive marking rates may be needed for satisfactory randomized congestion feedback under heavy traffic loads. We also study the 'TCP-friendliness' of our ECN-mod protocol by observing the potential unfairness in resource-sharing between conventional TCP and ECN-mod flows.

2 Generalized Congestion Control and ECN-mod

Consider an operating environment where an IP flow achieves reliable transmission by using per-packet acknowledgment. Whenever a router recognizes the onset of congestion in its buffer, it sets a "Congestion Experienced" (CE) bit (also called *marking* the packet) in the header of appropriate packets. By having the destination echo this bit in an acknowledgment packet, the source can be informed of such network congestion.

For window-based protocols operating under the TCP paradigm, source adaptation to such congestion can be described by the following generalized behavior: *Whenever an acknowledgment arrives for an unmarked data packet, the congestion window increases from its current value W by $incr(W)$. If, however, the acknowledgment indicates that the data packet had been marked in the forward path, the congestion window is decreased from W by $decr(W)$.*

Note that our framework is much simpler than alternative congestion control models suggested for the Internet (e.g., [8, 9]), which are directly concerned with ensuring fairness among competing flows. For example, [8] proposed a rate-based algorithm, where links explicitly update and propagate their shadow congestion costs, and where a source directly adjusts its rate based on its own cost sensitivity. [10] presented the Random Early Marking (REM) algorithm where such shadow costs could be communicated simply by intelligently adjusting the packet marking probability in the network buffers; sources in that scheme, however, adjust their congestion window only periodically. On the other hand, [9] proposed Charge-Sensitive TCP, where each flow needs to be aware of its instantaneous round-trip delay, transmission rate and congestion window size, and then uses an explicit target window size to regulate the growth of the congestion window. In contrast, our framework does not assume such intelligence at the TCP source, and does not necessarily require the network buffers to dynamically adjust their packet marking function.

For a constant marking probability p, the 'drift' (or the change in the expected value of the congestion window W_{n+1} at the $(n+1)^{th}$ acknowledgment, given the window W_n after the n^{th} acknowledgment) is given by

$$E[W_{n+1} - W_n | W_n = W] = drift(W, p) = (1-p).incr(W) - p.decr(W) = p.incr(W).\left(\frac{1-p}{p} - \frac{decr(W)}{incr(W)}\right). \tag{1}$$

Let $q(W)$ be the function

$$q(W) = \frac{decr(W)}{incr(W)}. \tag{2}$$

In that case, if the marking probability p is constant, $q(W)$ will fluctuate around $\frac{1-p}{p}$. The function $q(W)$ is really the *response surface* of the sources to router behavior; as a protocol designer, we can thus first choose $q(.)$ arbitrarily, and then still choose between different values of $incr(.)$ and $decr(.)$ (as long as their ratio remains unchanged). In fact, a legitimate way of designing a congestion protocol is to first choose the response surface $q(W)$ and then directly adapt the window W from an estimate of the marking probability p, without even defining separate $incr(.)$ and $decr(.)$ functions (an approach used in [11]).

For practical reasons, we restrict ourselves to the 'polynomial' class [5] of adaptation algorithms, where

$$incr(w) = c_1 w^{\alpha},\ decr(w) = c_2 w^{\beta}. \tag{3}$$

To ensure that the window does not grow without bound for any given probability, we need $\alpha < \beta$. For the polynomial class of algorithms, this drift would be 0 when $(1-p) * c_1 * W^{\alpha} = p * c_2 * W^{\beta}$. Accordingly, a flow transporting a very large file and subject to a constant marking probability p would observe its congestion window fluctuate around a central value $w(p)$, given by:

$$w(p) = \left(\frac{c_1}{c_2}\frac{1-p}{p}\right)^{\frac{1}{\beta-\alpha}}. \tag{4}$$

2.1 Current TCP Response and Our ECN-mod Algorithm

Under TCP's current *congestion avoidance* algorithm [13], the congestion window *cwnd* (expressed in terms of the MSS or Maximum Segment Size) increases 1 once every round trip time (RTT) in the absence of congestion; on detection of a congestion episode, *cwnd* decreases from its instantaneous value W by $\frac{W}{2}$. Neglecting transients such as fast recovery and slow-start, TCP's congestion control mechanism is thus a member of the polynomial class, with the parameters

$$TCP:\ c_1 = 1,\ \alpha = -1,\ c_2 = \frac{1}{2},\ \beta = 1. \tag{5}$$

Of course, most modern TCP versions, such as NewReno or Vegas, halve their window only once for multiple packet losses occurring within a single window (and thus presumably corresponding to a single congestion event).

To provide a more reactive ECN-mod TCP, we use two modifications:

- Make $incr(W)$ more aggressive than TCP, so that it can rapidly increase its *cwnd* in the absence of marking.
- Make $decr(W)$ milder, so that an ECN-mod source can reduce its sending rate in a much more gradual manner.

Of course, to throttle sources rapidly in such an environment, the marking probability for ECN-mod sources should be corresponding higher; more precisely, the buffers should have a higher slope in the marking function. Moreover, our ECN-mod protocol is assumed to respond to *all* ECN-marked packets, even if it leads to multiple reductions within a single window worth of packets. The detailed analysis for the specific choices for β, α, c_1 and c_2 was presented in [3, 4], which also recommended an implementation-friendly version of ECN-mod with $\beta = 1, \alpha = 0$.

Since c_1 and c_2 are scaling constants, their choice was more a matter of proper engineering design. We merely require c_2 to be smaller than $\frac{1}{2}$ (current TCP practice) to achieve milder backoff and c_2 to be corresponding small to have reasonable values for the 'average' window size ($w(p)$ in equation (4)) for moderately small p. We have thus experimentally studied a variety the the following members of the ECN-mod family of algorithms

$$ECN - mod: \ c_1 = \{0.625, 0.025\}, \ \alpha = 0, \ c_2 = \frac{1}{8}, \ \beta = 1, \tag{6}$$

which ensure that, for small p, the expected number of marked packets per RTT, lies in the range $(\frac{1}{5}, 5)$.

3 Simulation Parameters and Choices

Our simulation studies are performed using the ns-2 [14] simulator. To simulate a variable-bandwidth environment for best-effort traffic, we used Voice-over-IP (VoIP) sources as higher priority traffic. While each VoIP flow was modeled as per the specifications of the G.711 codec as an exponentially modulated on-off process, the total number of instantaneous calls was modeled as a birth-death process, with call arrival rate λ and exponentially distributed holding times with mean $\frac{1}{\mu}$.

For the graphs plotted here, the best-effort flows were either

- "ECN-aware NewReno", which implements the current TCP algorithm of halving the window in response to both dropped and marked packets.
- "ECN-mod" (or modified ECN), where the source reacts to marked packets as in section II.B and to dropped packets as in TCP NewReno.

We used both a) persistent TCP sources, which involved the transfer of infinite-sized files, and b) Web-TCP sources (using parameters reported in [15]), where a single flow alternates between a *active transfer* phase (during which a new TCP connection is used to transfer a finite-sized file) and an *inactive* phase where the source remains in an idle state.

3.1 Router Marking/Dropping Behavior

Random packet marking and dropping was implemented by a RED [12] queue. The marking function (for ECN NewReno flows), $p(Q)$, was based on the 'gentle' variant [16] of RED and is denoted as $p(Q)$, with the marking probability a linear function of the queue occupancy Q. For ECN-mod flows, the marking function was modified to be more aggressive, such that: *given a queue occupancy Q, the average congestion window size for a best-effort flow was the same for all choices of the congestion window protocol.* Accordingly, the marking function for ECN-mod packets, $p_{mod}(Q)$ is:

$$p_{mod}(Q) = \left(1 + \frac{c_2}{c_1} * \sqrt{\frac{2 * (1 - p(Q))}{p(Q)}}\right)^{-1}, \tag{7}$$

where $p(Q)$ is the basic RED marking function.

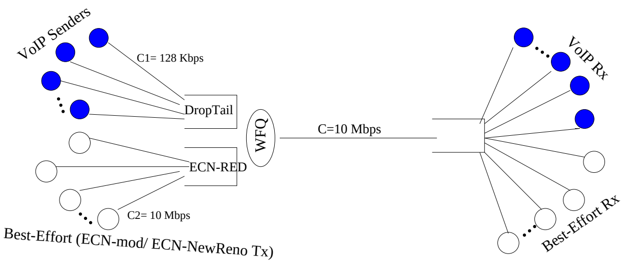

Figure 1: Simulation Topology for WFQ Experiments

4 Effectiveness and Parameter Insensitivity of ECN-mod

We now report on simulation studies that investigate how ECN-mod flows perform relative to ECN-NewReno flows, and how their utilization varies with changes in the offered load and marking probabilities. The simulation topology is as shown in Figure 1, with higher-priority (VoIP) and best-effort (TCP) traffic buffered in two separate queues, and Weighted Fair Queuing (more precisely, SCFQ) used to isolate the two classes. To provide voice higher priority, the VoIP class had a weight of 0.8, compared to 0.2 for TCP traffic, even though the offered load of VoIP traffic was often much lower than that of TCP. Admission control is performed for VoIP traffic by having the network block more than $Max.VoIP$ simultaneous voice sessions. For the plots provided here, the bottleneck link capacity C is 10 Mbps; the VoIP queue was sized to have a maximum drain time of 20 msecs. The RED parameters for the best-effort queue (in packets) had $min_{th} = 25$, $max_{th} = 75$ and buffer size $B = 150$ (following the recommendations in [16]). The RTT of the best-effort connections are uniformly spaced out over the interval $(25, \ldots, 250)$ msecs.

4.1 Performance Improvement with Web TCP Traffic

The effect of ECN-mod in increasing the link utilization and TCP throughput for persistent TCP traffic was presented in [4]. In this subsection, we thus focus on the network performance when the sources are not persistent but rather represent finite-sized Web-based file transfers (using the Barford-Crovella model). Figures 2 and 3 plot the simulation results (averaged over 10 runs) when N, the number of Web TCP sources, equals 150 and $p_{max} = 0.2$. Figure 2 plots the total goodput (VoIP+ TCP), as well as the TCP goodput alone as the average number of simultaneous VoIP calls is varied (by varying λ). It is easy to see that ECN-mod and ECN-NewReno do not exhibit significant differences, although ECN-mod (for well-chosen values of c_1) does achieve slightly better utilization than ECN-NewReno. The reason for this is easy to understand: while the dynamic variation in the number of active TCP transactions will cause transient network congestion, the long-term TCP throughput does not change since the best-effort traffic is essentially source-constrained. More importantly, unlike earlier studies with persistent TCP traffic, setting $c_1 = 0.125$ in ECN-mod performs better than $c_1 = 0.0625$. Since most Web file transfers usually complete during the initial slow-start transient (before congestion feedback is even activated), a more aggressive choice of the window increase coefficient typically leads to higher TCP goodput. Of course, as in [4], an over-aggressive value of c_1 can increase the queue variability significantly, leading to buffer underflow and loss of network utilization[1].

Figure 3 plots the packet marking rates for the best-effort flows and the coefficient of variation (defined as $\frac{Std.Deviation}{Mean}$) of the best-effort queue occupancy. As expected, the marking rates turn out to be higher for ECN-mod than ECN-NewReno– our congestion control framework is based on a more aggressive congestion notification ability in Internet routers. (The packet loss rate on all these runs was essentially 0, indicating that congestion control was achieved solely via packet marking). More importantly, the coefficient of variation for ECN-mod flows is lower (sometimes by as much as 30%) than that of ECN-NewReno flows. Since ECN-mod flows exhibit a much milder backoff than ECN NewReno, the occupancy of the RED queue (for good choices of ECN-mod parameters) fluctuates in a much smoother fashion, leading to much smaller coefficients of variation than that with ECN-NewReno. Accordingly, while the use of the ECN-mod window adjustment protocol may not improve the long-term

---

[1] This result suggests an interesting possibility of having c_1 during the initial slow-start transient different from the subsequent value of c_1 during the congestion avoidance phase; we do not, however, explore this idea further in this paper.

network utilization significantly (since the Web traffic load is essentially source-constrained), it does lead to better the network dynamics, such as a smoother queue evolution and lower packet jitter.

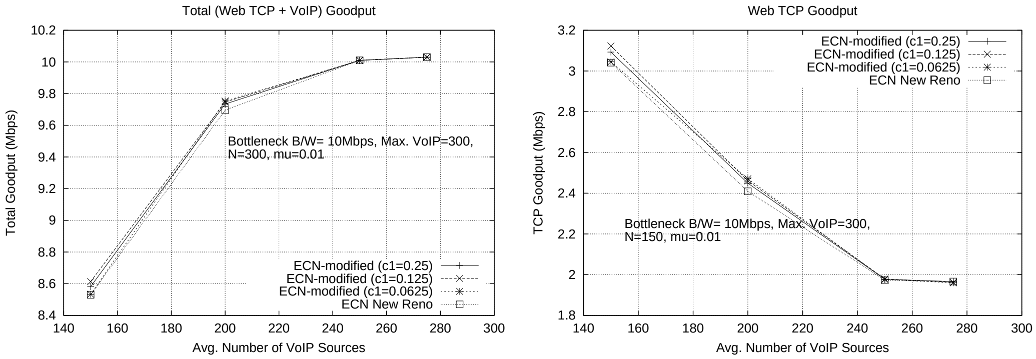

Figure 2: Comparative Capacity Utilization for Web Sources

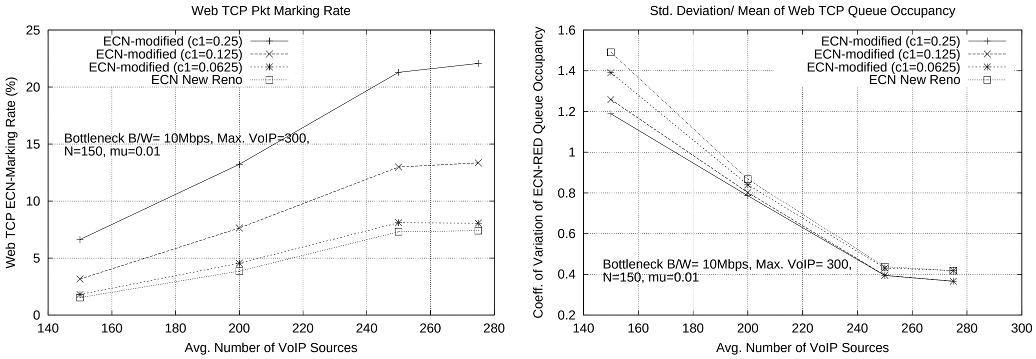

Figure 3: TCP Marking Rates and Queue Variability for Web Sources

4.2 Sensitivity to Load Variation

We now consider the performance of ECN-mod vs. ECN-NewReno as the number of *persistent* best-effort flows is varied. (Results with Web TCP sources are qualitatively similar and omitted due to space constraints.) For these studies, the average number of VoIP flows was kept constant at 200, by setting $\lambda = 2.0$. We varied the number of best-effort (persistent) flows, N, from $10 - 200$. The graphs study two interesting settings of p_{max}, namely 0.2 and 1.0.

We first consider the commonly used RED setting of $p_{max} = 0.2$. Figure 4 shows the variation in the total (VoIP+best-effort), as well as the best-effort, goodput as N is varied from 10 to 200. We see that, when N is relatively large, both ECN-mod and ECN-NewReno obtain comparable goodput. However, when

N is small, ECN-mod ($c_1 = 0.0625$) clearly outperforms ECN-NewReno, since ECN-mod is able to utilize the available bandwidth more aggressively. Figure 5 shows the variation in the packet dropping and marking rates respectively. We see that, as N increases, the packet loss rates become very high (as large as $\sim 10\%$ for $N = 200$). This indicates that a p_{max} setting of 0.2 does not provide sufficiently strong feedback to prevent undesirable packet losses under high loads– a larger value of p_{max} is preferred.

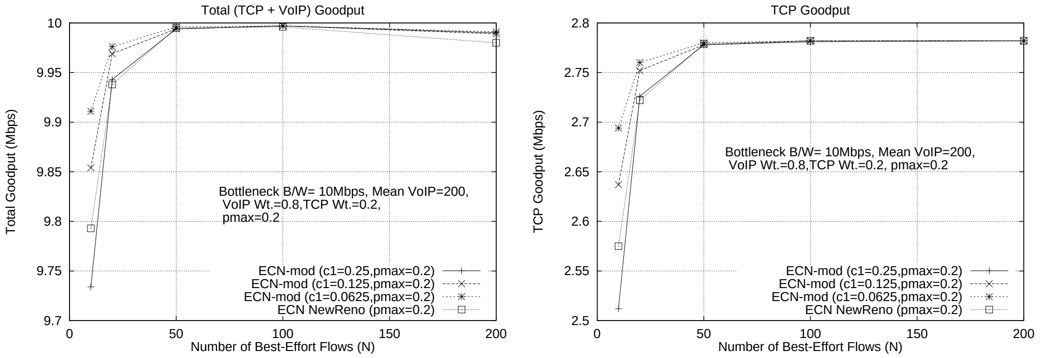

Figure 4: Comparative Goodput ($p_{max} = 0.2$)

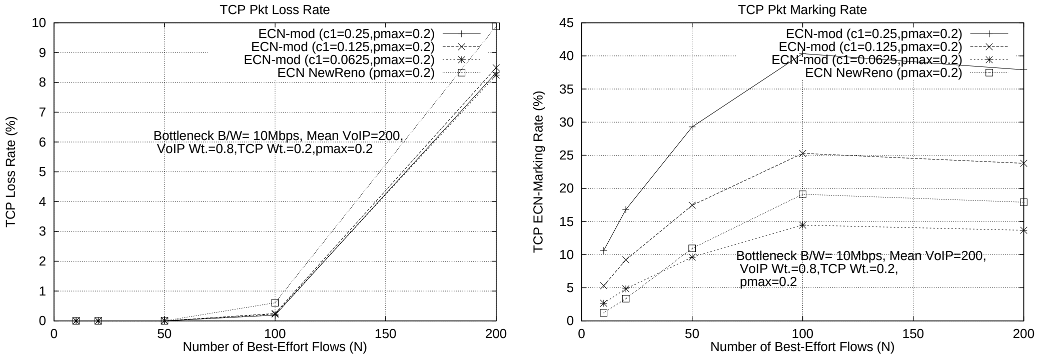

Figure 5: Comparative Drop/Marking Rates ($p_{max} = 0.2$)

In Figures 6 and 7, we investigate precisely such an aggressive setting, where $p_{max} = 1$. Figure 6 shows the total and best-effort traffic goodput as N is varied. As before, we see that ECN-mod is better than ECN-NewReno in utilizing the available bandwidth. More importantly, as N is increased beyond 50, we see that, while the ECN-mod flows ($c_1 = 0.0625$) always achieve high goodput, the ECN-NewReno goodput actually decreases. This occurs because TCP's policy of halving the window size does not work well at the relatively high marking rates (see Figure 7) obtained when $p_{max} = 1.0$ and N is large. Thus, *the current* *response of TCP to ECN marking does not allow best-effort flows to operate in environments where routers exhibit aggressive marking behavior.*

The first graph in Figure 7 plots the average marking rates for best-effort traffic, when $p_{max} = 1.0$. Observe that the marking rates for ECN-mod traffic are as high as $\sim 85\%$; yet, Figure 6 shows no degradation in ECN-mod performance. The second graph plots the coefficient of variation of the queue occupancy as N is increased. While ECN-mod always results in a lower queue variability (smaller coefficient of variation) than ECN-NewReno, the difference is more pronounced for large N, where ECN-NewReno cannot cope with the high marking rates. It is well-known that RED's inability to adaptively vary p_{max} leads to performance degradation as the number of TCP flows is varied. Accordingly, ECN-mod appears to provide the significant advantage of making the best-effort utilization largely independent of the number of active flows; by setting p_{max} to a high value and using the ECN-mod algorithm, we can make the network utilization uniformly high for a very wide range of N and avoid undesirable packet drops.

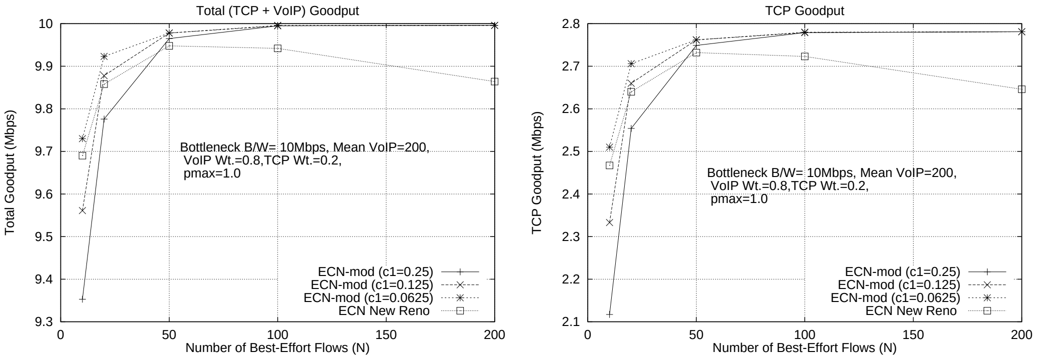

Figure 6: Comparative Goodput ($p_{max=1.0}$)

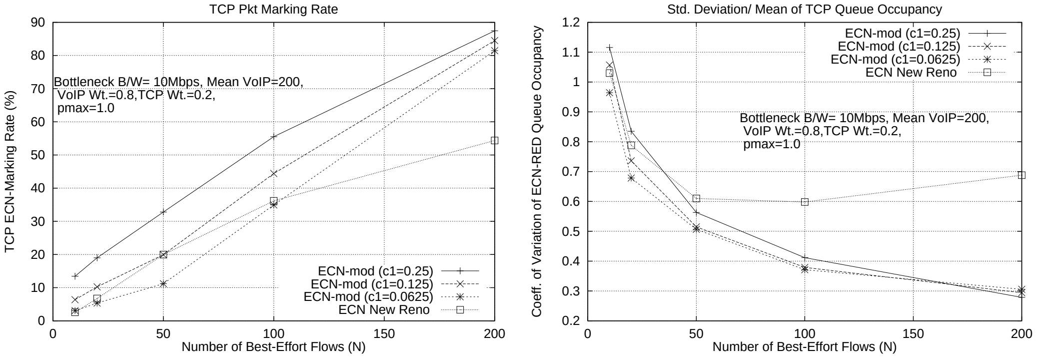

Figure 7: Comparative Marking/Queue Variation ($p_{max=1.0}$)

5 Fairness between ECN-mod and TCP NewReno

Since it is impractical to expect that all sources to change their window-adjustment behavior overnight, we have also studied the "TCP-friendliness" of ECN-mod traffic, i.e., the relative sharing of the best-effort bandwidth between competing ECN-mod and conventional TCP flows. To this end, we performed simulations where the router port uses a single FIFO buffer, and the best-effort flows were either all NewReno or ECN-mod or an equal mix of both. The network topology is thus similar to that of Figure 1, except that VoIP no longer has explicit protection through the Class Based WFQ mechanism. For these experiments, the link capacity $C = 10$ Mbps, $min_{th} = 20$, $max_{th} = 60$, $p_{max} = 0.2$ and buffer size $B = 120$. As before, due to space limitations, we report only on experiments with persistent TCP sources.

Figure 8 shows the variation in the total goodput, as well as the TCP goodput, for the various TCP adaptation algorithms when the total number of best-effort sources, N, equals 20. We can clearly see that ECN-mod, with $c_1 = 0.0625$, outperforms the current ECN-NewReno procedure. Once again, we observe that ECN-mod with $c_1 = 0.25$ performs worse than the current ECN-NewReno algorithm, indicating that an overly aggressive choice of parameters may incur severe performance penalties. We also observed that the VoIP throughput was essentially unaffected by the best-effort traffic, since the non-adaptive UDP flows do not react to the 'marking' of packets at the buffer.

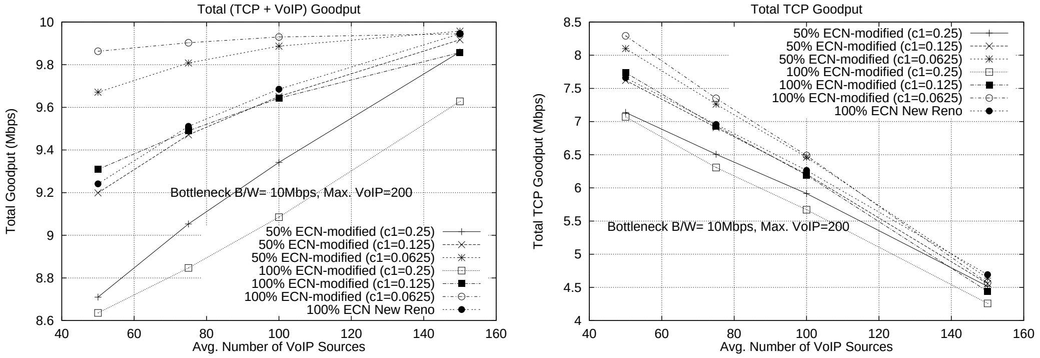

Figure 8: Throughput for Mixed NewReno/ECN-mod Traffic

Figure 9 first plots the relative throughputs of the ECN-mod and ECN-NewReno flows, when the best-effort traffic consists of an equal number (10 each) of ECN-mod and ECN-NewReno sources. Clearly, while ECN-mod has the higher goodput (for $c_1 = 0.0625$), ECN-NewReno sources are not completely shut out and obtain about $20\% - 25\%$ less goodput than their ECN-mod counterparts. A far more important point can be observed by studying the second graph, which studies the goodput achieved by the 10 ECN-NewReno sources, when the other 10 sources were either ECN-NewReno or ECN-mod. It is interesting to see that, for certain loads, the 10 ECN–NewReno sources obtain higher goodput if the other sources are ECN–mod ($c_1 = 0.0625$) than if they are ECN–NewReno. This illustrates the important point that, under certain circumstances, *the performance of conventional ECN-NewReno sources is improved (in absolute terms) in the presence of other ECN-mod traffic sources, even though, relatively speaking, the ECN-NewReno sources receive a smaller fraction of the total goodput.* This clearly mitigates any potential fairness concern, since the overall increase in the utilization levels swamps the reduction in ECN-NewReno's share of the total achieved goodput. Additional plots (omitted here due to space constraints) further show that ECN-mod (with $c_1 = 0.0625$) flows result in a lower coefficient of variation of the queue occupancy than corresponding ECN-NewReno flows. In this case, where VoIP packets are buffered in the same queue, this directly translates into *smaller delay jitter* for individual VoIP packets.

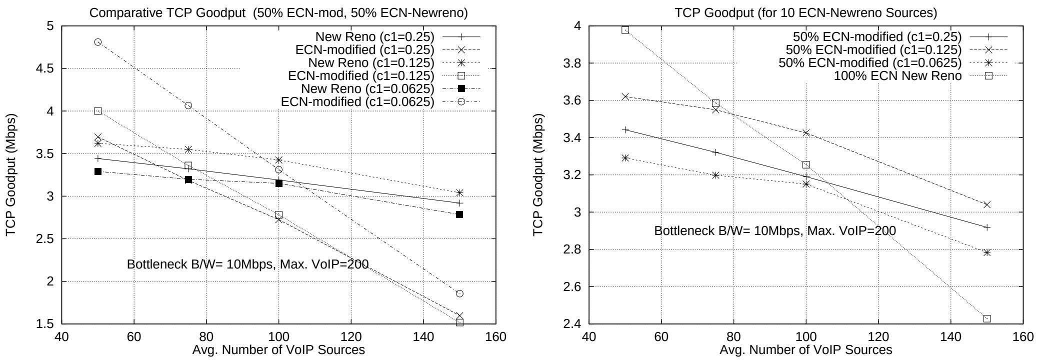

Figure 9: Relative Goodput for ECN-NewReno and ECN-mod Sources

We have also studied the fairness issues between ECN-mod and non-ECN capable TCP flows using the same setup. As expected, ECN-unaware flows perform worse than ECN-capable ECN-mod flows. However, those studies also demonstrate that ECN-mod, while capturing a relatively larger portion of the available bandwidth, *never leads to starvation or significant deterioration* of the conventional TCP flows. Such studies indicate that it may be possible for ECN-mod and conventional TCP flows to co-exist in the network, especially if the network buffers are able to apply a more aggressive marking behavior selectively to the ECN-mod flows.

6 Conclusions

In this paper, we continue our investigation of a rapidly reactive congestion control framework for adaptive (best-effort) TCP-like flows. This framework includes an ECN-mod protocol that has a more aggressive decrease and milder decrease than conventional TCP, and requires routers to mark packets much more aggressively than currently envisioned. Simulation studies indicate the performance benefits of ECN-mod over ECN-NewReno, demonstrated earlier for persistent TCP sources, apply even when the flows transfer finite-sized files and are source-constrained. In particular, the use of ECN-mod window adaptation leads to smoother buffer behavior and less drastic variation in the instantaneous total traffic loads. We, however, need to be conservative in the choice of ECN-mod coefficients: if the window increase coefficient is too large, network utilization may drop significantly.

Further studies also show that the use of the ECN-mod protocol makes the link utilization by adaptive traffic significantly less sensitive to the number of active flows, and the precise setting of RED's p_{max} parameter. Studies using a mixture of ECN-mod flows and conventional TCP flows also demonstrate that ECN-mod does not significantly penalize conventional TCP traffic; while ECN-mod does grab a larger share of the available bandwidth, it also improves the overall utilization. While not intended to be conclusive, our results do argue that the current TCP behavior, of responding to the notification of an ECN-marked packet in exactly the same way as it reacts to the discovery of a lost packet ([1, 2]), may be sub-optimal. The best shape of the marking function, however, remains an open question.

References

1. Floyd, S.: TCP and Explicit Congestion Notification, *ACM Computer Communications Review* 21 no 5, 1994.
2. Ramakrishnan, K.K., Floyd, S.: A Proposal to add Explicit Congestion Notification(ECN) to IP, *RFC 2481*, January 1999.
3. Ott, T.J.: ECN Protocols and the TCP Paradigm, *ftp://ftp.research. telcordia.com/pub/tjo/ECN.ps*, 1999.
4. Misra, A., Ott, T.J.: Jointly Coordinating ECN and TCP for Rapid Adaptation to Varying Bandwidth, *Proceedings of IEEE MILCOM 2001*, to appear, October 2001.
5. Bansal, D., Balakrishnan, H.: Binomial Congestion Control Algorithms, *Proceedings of IEEE INFOCOM*, April 2001.
6. Ott, T., Lakshman, S., Wong, L.: SRED: Stabilized RED, *Proceedings of IEEE INFOCOM*, March 1999.
7. Feng, W., Kandlur, D., Saha, D. and Shin, K.: Adaptive Packet Marking for Providing Differentiated Services in the Internet, *Proceedings of ICNP'98*, October 1998.
8. Kelly, F.P., Maulloo A.K., Tan D.K.H.: Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability, *Journal of the Operational Research Society*, March 1998.
9. La, R.J., Anantharam V.: Charge-Sensitive TCP and Rate Control in the Internet, *Proceedings of IEEE INFOCOM*, March 2000.
10. Low, S.H., Lapsley D.E.: Optimization Flow Control, *IEEE/ACM Transactions on Networking*, December 1999.
11. Floyd, S., Handley M., Padhye J., Widmer J: Equation-based Congestion Control for Unicast Applications, *Proceedings of ACM SIGCOMM 2000*, Sept 2000.
12. Floyd, S., Jacobson V., Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, August 1993.
13. Jacobson, V., Karels. K.: Congestion Avoidance and Control, *Proceedings of ACM SIGCOMM*, 1988.
14. The ns-2 network simulator, *http://www-mash.CS.Berkeley.EDU/ ns.*
15. Barford M., Crovella M.: Generating Representative Workloads for Network and Server Performance Evaluation, *Boston University Technical Report*, BU-CS-97-006, 1997.
16. Floyd, S.: Recommendations on using the 'gentle' variant of RED, *http://www.aciri.org /floyd/red.html,* March 2000.