A Performance Study of Explicit Congestion Notification (ECN) with Heterogeneous TCP Flows

Robert Kinicki and Zici Zheng

Worcester Polytechnic Institute
Computer Science Department
Worcester, MA 01619

Abstract. This paper compares the simulated performance of RED routers and ECN routers. The results show that ECN provides better goodput and fairness than RED for heterogeneous flows. When the demand is held constant, the number of flows generating the demand has a negative effect on performance. ns-2 simulations with many flows demonstrate that the bottleneck router's marking probability must be aggressively increased to provide good ECN performance. These experiments suggest that an adaptive version of ECN should provide better performance than ECN.

1 Introduction

With increased World Wide Web traffic has come heightened concern about Internet congestion collapse. Since the first congestion collapse episode in 1986, several variants of TCP (Tahoe, Vegas, Reno and NewReno) have been developed and evaluated to provide host-centric mechanisms to combat high packet loss rates during heavy congestion periods. Additionally, researchers have proposed new congestion avoidance techniques for Internet routers. While the initial concept was to use packet loss at FIFO routers to signal congestion to the source, the resulting drop-tail behavior failed to provide adequate early congestion notification and produced bursts of packet drops that contribute to unfair service.

Since the introduction of Random Early Detection (RED) [6] in 1993, researchers have proposed a variety of enhancements and changes to router management to improve congestion control while providing fair, best-effort service. Although RED has outperformed drop-tail routers in several simulation and test-bed experiments [1], [4], [5], [8], [9], [12], Christainsen et al [3] have demonstrated that tuning RED for high performance is problematic when one considers the variability of Internet traffic.

RED has been shown to be unfair when faced with heterogeneous flows [10] and the recommended RED parameter settings are not aggressive enough in heavy congestion generated by a large number of flows [3], [5], [8].

Concern over reduced performance on the Internet during traffic bursts such as Web flash crowds helped spawn the IETF recommendation [2] for new active queue management techniques that provide early congestion notification to TCP sources. Several research studies [1], [7], [8], [9], [15] have reported better performance for Explicit Congestion Notification (ECN) when compared against RED. These results add support to the Internet draft "Addition of ECN to IP" [14]. However, most of these studies cover only a limited portion of the traffic domain space. Specifically, little attention has been given to evaluating the effects of a large number of concurrent flows. Although a couple of these studies consider fairness among competing homogeneous flows, ECN behavior with heterogeneous flows has not been thoroughly studied.

This paper presents results from a series of ns-2 simulations comparing the ability of RED and ECN to provide fair treatment to heterogeneous flows. The goal of this report is to add to the existing information on ECN behavior specifically with regard to the impact of the number of flows, the effect of ECN tuning parameters on performance, and the effectiveness of ECN's congestion warnings when many flows cause the congestion. The results of this study provide insight into a new active queue management scheme, AECN, Adaptive ECN.

Section 2 briefly defines a few measurement terms and reviews previous ECN studies to provide context for our experiments. Section 3 discusses experimental methods. The next section analyzes the simulated results and the final section includes concluding remarks.

2 Definitions and Background

The performance metrics used in this investigation include delay, goodput and two ways to evaluate fairness. The delay is the time in transit from source to destination and includes queuing time at the router. Goodput differs from throughput in that it does not include retransmitted packets in the count of packets successfully arriving at the receiver. Given a set of flow throughputs

$$(x_1, x_2, \ldots, x_n)$$

Jain's fairness index [13] is defined in terms of the following function

$$f(x_1, x_2, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} (x_i)^2}$$

A second form of fairness introduced in section 4 focuses on the difference between the maximum and minimum average goodput for groups of heterogeneous flows [8].

Random Early Detection (RED) [6] utilizes two thresholds (*min_th*, *max_th*) and an exponentially-weighted average queue size, *ave_q*, to add a probabilistic drop region to FIFO routers. *max_p* is a RED tuning parameter used to control the RED drop probability when *ave_q* is in the drop region. The drop probability increases linearly towards *max_p* as *ave_q* moves from *min_th* to *max_th*. When *ave_q* reaches *max_th*, RED switches to a deterministic (100%) drop probability. *max_th* is set below the actual queue length to guarantee drops that signal router congestion before the physical queue overflows.

Explicit Congestion Notification (ECN) [12],[14] marks a packet (instead of dropping) when *ave_q* is in the probabilistic drop region. In the deterministic drop region, ECN drops packets just as RED does. We breifly consider an ECN variant, ECNM, that marks packets in the deterministic region.

Lin and Morris [10] define fragile TCP flows as those eminating from sources with either large round-trip delays or small send window sizes and robust flows as having either short round-trip delays or large send windows. This delineation emphasizes a flow's ability to react to indications of both increased and decreased congestion at the bottleneck router. Our experiments simulate three distinct flow groups (fragile, average, and robust flows). These flows differ only in their end-to-end round-trip times (RTTs). The maximum sender window is held fixed at 30 packets in all graphs discussed in section 4 to simplify the analysis.

Floyd's original ECN paper [7] shows the advantages of ECN over RED using both LAN and WAN scenarios with a small number of flows. Bagal et al [1] compare the behavior of RED, ECN and a TCP rate-based control mechanism using traffic scenarios that include 10 heterogeneous flows. They conclude that RED and ECN provide unfair treatment when faced with either variances due to the RTTs of the heterogeneous flows or variances in actual flow drop probabilities. Focusing on a window advertising scheme (GWA), Gerla et al [8] compare GWA, RED, and ECN in scenarios with up to 100 concurrent flows. Using the gap between maximum and minimum goodput as a fairness measure, they show that ECN yields better fairness than RED for homogeneous flows. Salim and Ahmed [16] use Jain's fairness to compare ECN and RED performance for a small number of flows. Their results emphasize that *max_p* can significantly effect performance. The ns-2 experiments discussed in this paper combine and extend these results.

3 Experimental Methods and Simulation Topology

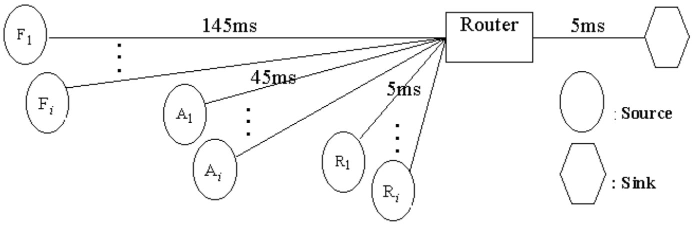

Fig. 1: Simulation Topology

This study uses the newest version of Network Simulator from UCB/LBNL, **ns-2** [11], to compare the performance of ECN and RED routers with TCP Reno sources. The simulation network topology (shown in Figure 1) consists of one router, one sink and a number of sources. Each source has a FTP connection feeding 1000-byte packets into a single congested link. The bandwidth of the bottleneck link is 10Mbps with a 5 ms delay time to the sink. The one-way link delays for the fragile, average and robust sources are 145 ms, 45 ms and 5 ms respectively. Thus, the fragile, average and robust flows have round-trip times of 300 ms, 100 ms and 20 ms when there is no queuing delay at the router.

All simulations ran for 100 simulated seconds. Half the flows were started at time 0 and the other half were started at 2 seconds. The graphs presented exclude the first 20 seconds to reduce transient startup effects. The router for all simulations have a *min_th* of 5 packets and a physical queue length of 50 packets. Except for the maximum send window size of 30 packets, all other parameters use the ns-2 default values.

4 Results and Analysis

A series of ns-2 experiments were run such that the cumulative traffic flow into the heavily congestion router **remains fixed** at 600 Mbps even though the number of flows is varied across simulations. In all cases, the number of flows is equally divided among the three flow categories. Thus, 15 flows in the graphs implies 5 fragile, 5 average and 5 robust flows each with a 40 Mbps data rate whereas a graph point for 120 flows implies a simulation with 40 fragile, 40 average and 40 robust flows each with a 5 Mbps data rate. Simulations were run with the total number of flows set at 15, 30, 60, 120, 240, 480 and 600 flows.

Figure 2 gives ECN and RED goodput with the number of flows varying from 15 to 600. ECN with $max\_p = 0.5$ provides the best goodput in all cases except 15 flows. In the other three cases there is a marked drop in goodput beginning at 64 flows. Figure 3 presents the delay for ECN and RED with $max\_p = 0.5$. This figure shows the clear advantage robust flows have with respect to delay, but more importantly it demonstrates that the ECN goodput improvement from Figure 2 is offset by a small increase in the one-way delay for ECN.

Figures 4 and 5 track the effect of varying *max_p* and *max_th* in simulations with 30 and 120 flows respectively. Figure 4 shows that *max_th* has little effect on goodput above $max\_p = 0.2$. In Figure 5 where 120 flows provide the same flow demand as 30 flows in Figure 3, ECN with $max\_p = 0.5$ and $max\_th = 30$ yields the highest goodput and there is no *max_p* setting for RED that works well.

Figure 6 employs Jain's fairness to quantify RED and ECN behavior. ECN is fairer than RED in almost all situations.

Since perfect fairness has a Jain's fairness index of 1, it is clear that as the number of flows goes above 120 none of the choices prevent unfairness. The fact that ECN with $max\_p = 0.1$ is fairest at 30 flows while $max\_p = 0.5$ is the fairest at 60 and 120 flows implies the marking probability should be dynamically

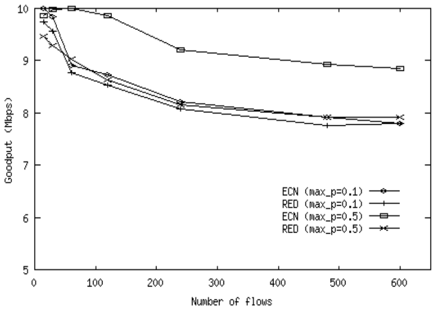

Fig. 2: RED and ECN Goodput, *max_th*=30

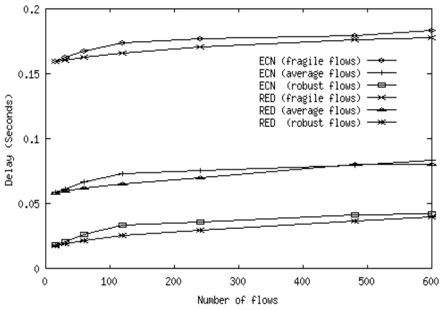

Fig. 3: RED and ECN Delay, *max_p*=0.5, *max_th*=30

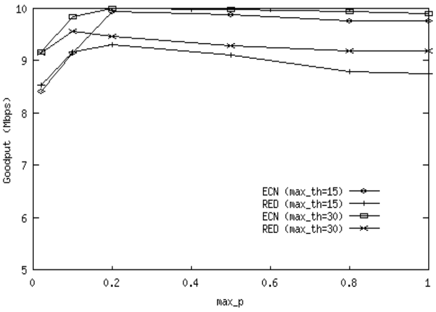

Fig. 4: Goodput with 30 flows

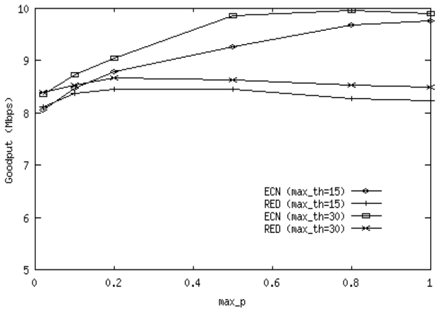

Fig. 5: Goodput with 120 flows

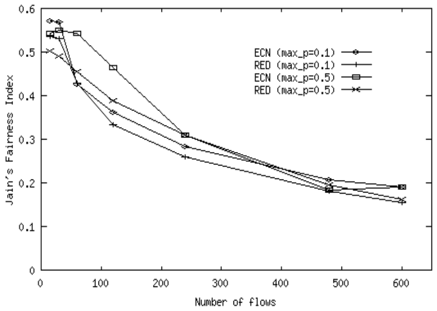

Fig. 6: RED and ECN Fairness (*max_th*=30)

adjusted based on a flow count estimator. The unfairness at a high number of flows can also be partially attributed to a lockout phenomenon where some flows are unable to get any flow through the congested router for the duration of the simulation. Locked out flows begin to appear for both RED and ECN above 120 flows.

Figures 7 through 9 provide a visual sense of max-min fairness via the gap between the averaged goodputs for the three flow groups.

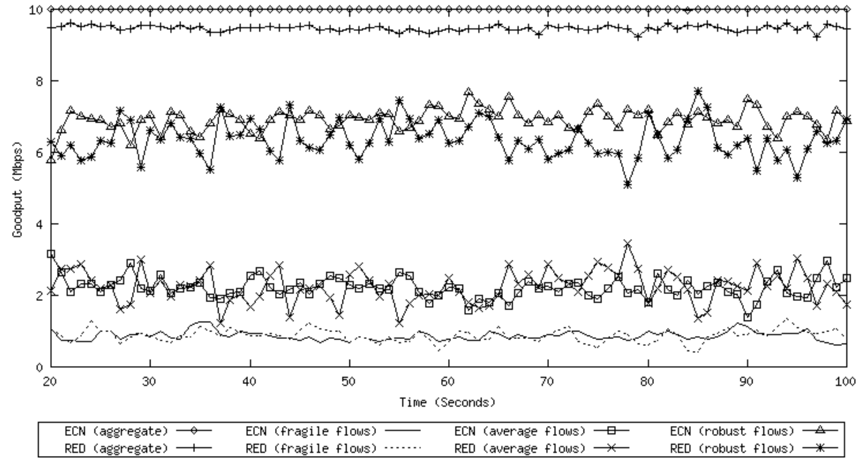

Fig. 7: Goodput Distribution, 30 flows, *max_p*=0.2, *max_th*=30

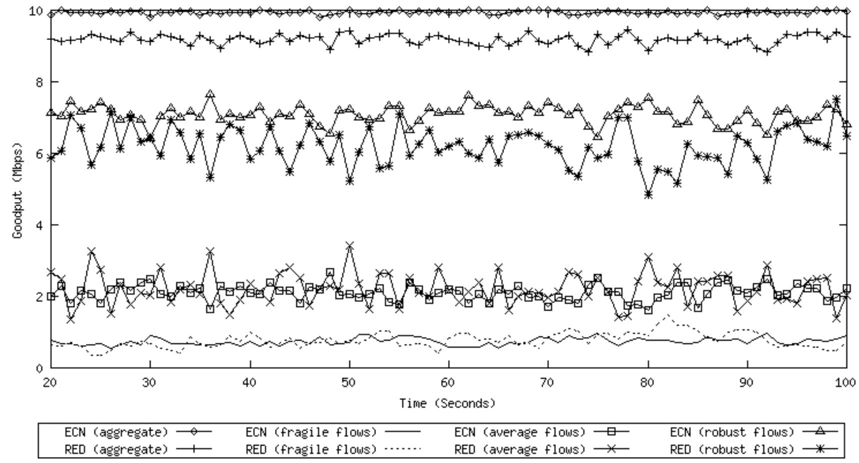

Fig. 8: Goodput Distribution, 30 flows, *max_p*=0.8, *max_th*=30

Aggregate goodput in these graphs is the sum of the fragile, average, and robust goodputs. ECN provides better overall goodput than RED in all three graphs, but the difference is most pronounced in Figure 9 where the traffic is generated by 120 flows. Figure 7 and 8 differ only in an increase of *max_p* from 0.2 to 0.8. The more aggressive ECN marking in Figure 8 provides better goodput for robust flows than RED. However this change does not reduce the goodput gap between robust and fragile flows. Figure 9 keeps *max_p* = 0.8 but simulates 120 flows. Although overall goodput remains relatively unchanged for ECN in Figure 9, the goodput for the robust flows goes down while the goodput of the average and fragile flows increase slightly. This implies that varying *max_p* when there are heterogeneous flows can provide improvement in the visual max-min goodput. RED goodput is adversely affected by more flows. This suggests an adaptive ECN that uses different values of *max_p* for the different flow groups.

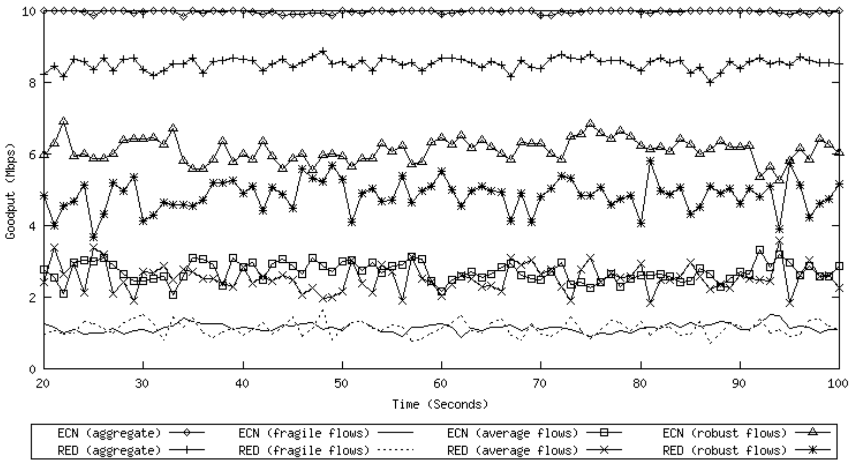

Fig. 9: Goodput Distribution, 120 flows, *max_p*=0.8, *max_th*=30

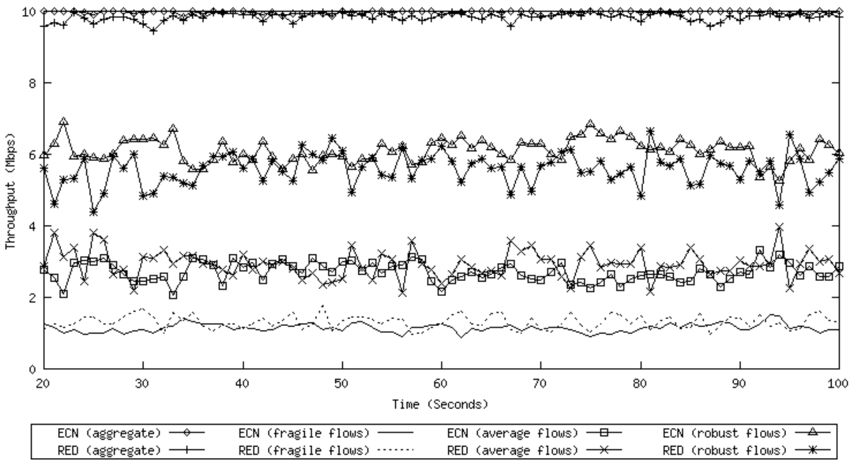

Fig. 10: Throughput Distribution, 120 flows, *max_p*=0.8, *max_th*=30

The significance of using goodput instead of throughput as a performance metric can be clearly seen in Figures 9 and 10. Because goodput excludes retransmissions, RED has 15% lower goodput than ECN in Figure 9. Since RED drops and ECN marks, the RED drops trigger more TCP retransmissions. This effect is completely hidden in Figure 10 where aggregate RED throughput is only slightly lower than aggregate ECN throughput.

Figure 11 compares ECN with ECNM. Recall ECNM differs from standard ECN in that ECNM marks packets when the average queue size exceeds *max_th* and drops packets only when the router queue overflows. The figure shows that ECN provides better goodput except at small values of *max_p* and that ECNM appears quite sensitive to the *max_th* setting.

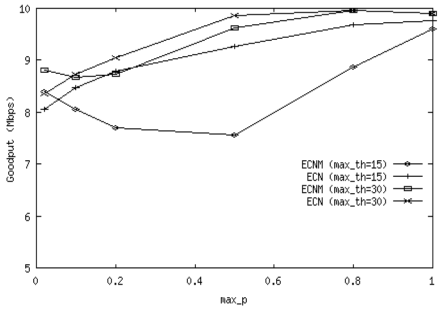

Fig. 11: ECN and ECNM Goodput with 120 flows

5 Conclusions and Future Work

This paper reports on a series of ns-2 simulations that compare ECN and RED performance with heterogeneous flows. Generally ECN provides better goodput and is fairer than RED. The results show that performance of both mechanisms are affected by the number of flows. However, ECN with an aggressive *max_p* setting provides significantly higher goodput when there are a large number of heterogeneous flows. ECN also had a higher Jain's fairness index in the range of flows just below where flow lockouts occurred.

In the simulations studied neither RED nor ECN strategy were fair to fragile and average flows. These results suggest that if congestion control is to handle Web traffic consisting of thousands of concurrent flows with some degree of fairness then further enhancements to ECN are needed. We are currently conducting simulations with an adaptive version of ECN that adjusts *max_p* based on the round-trip time of a flow and an estimate of the current number of flows in each flow groups.

References

1. P. Bagal, S. Kalyanaraman, B. Packer, Comparative study of RED, ECN and TCP Rate Control, Technical Report (Mar 1999).

2. B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D.Estrin, S. Floyd, V. Jacobson, G. Minshalll, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309 (Apr 1998).
3. M.Christainsen, K. Jeffay, D. Ott, F.D. Smith, Tuning RED for Web Traffic, Proc. SIGCOMM 2000, Stockholm (Sep 2000) 139-150.
4. W. Feng, D. Kandlur, D. Saha and K. Shin, A Self-Configuring RED Gateway, Proc. INFOCOM 1999, New York (Mar 1999) 1320-1328.
5. W. Feng, D. Kandlur, D. Saha and K. Shin, Blue: A New Class of Active Queue Management Algorithms, U. Michigan CSE-TR-387-99 (Apr 1999).
6. S. Floyd and V. Jacobson, Random Early Detection Gateways for Congestion Control, IEEE/ACM Transactions on Networking (Aug 1993) 397-413.
7. S. Floyd, TCP and Explicit Congestion Notification, ACM Computer Communication Review, **vol. 24 no. 5** (Oct 1994) 10-23.
8. M.Gerla, R.L.Cigno, S. Mascolo, W.Weng, Generalized Window Advertising for TCP Congestion Control, UCLA Tech Report 990012.
9. S. Kunniyur and R.Srikant, End-to-End Congestion Control schemes:Utility Functions, Random Losses and ECN Marks, Proceedings IEEE INFOCOM 2000, Tel-Aviv (Mar 2000) 1323-1332.
10. D. Lin and R. Morris, Dynamics of Random Early Dectection, ACM Computer Communication Review, **vol. 27, no. 4**, Cannes, France (Oct 1997) 127-138.
11. http://www.isi.edu/nsnam/ns/doc/index.html.
12. T. J. Ott, T. V. Lakshman, and L. H. Wong, SRED: Stabilized RED, Proceedings INFOCOM 1999, New York (Mar 1999) 1346-1355.
13. L.L. Peterson and B.S. Davie, Computer Networks, A Systems Approach, 2nd Ed., Morgan Kaufmann, San Francisco, 2000.
14. K.K. Ramakrishnan, and S. Floyd, A Proposal to add Explicit Congestion Notification (ECN) to IP, RFC 2481 (Jan 1999).
15. K.K. Ramakrishnan, S. Floyd, D. Black,The Addition of Explicit Congestion Notification (ECN) to IP, Internet Draft (Nov 2000).
16. J. H. Salim and U. Ahmed, Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks, RFC 2884 (Jul 2000).