IEEE GlobeCom (GLOBECOM 2002), Taipei, Taiwan, November 2002. LA-UR 02-5524

GREEN: Proactive Queue Management over a Best-Effort Network

Wu-chun Feng†, Apu Kapadia‡†, Sunil Thulasidasan†
feng@lanl.gov, akapadia@uiuc.edu, sunil@lanl.gov

† Research & Development in Advanced Network Technology (RADIANT)
Computer & Computational Sciences Division
Los Alamos National Laboratory
Los Alamos, NM 87545

‡ Department of Computer Science
University of Illinois at Urbana-Champaign
1304 W. Springfield Ave.
Urbana, IL 61801

***Abstract*— We present a proactive queue-management (PQM) algorithm called GREEN that applies knowledge of the steady-state behavior of TCP connections to intelligently and proactively drop packets, thus preventing congestion from ever occurring and ensuring a higher degree of fairness between flows. This congestion-prevention approach is in contrast to the congestion-avoidance approach of traditional active queue-management (AQM) schemes where congestion is actively detected early and then reacted to.**

In addition to enhancing fairness, GREEN keeps packet-queue lengths relatively low and reduces bandwidth and latency jitter. These characteristics are particularly beneficial to real-time multimedia applications. Further, GREEN achieves the above while maintaining high link utilization and low packet loss.

I. Introduction

Because network congestion leads to lost packets, thus wasting all the resources that the packet consumed on its way from source to destination, active queue-management (AQM) schemes such as RED [1] and Blue [2] have been proposed to actively detect congestion early and appropriately react to the impending congestion that would otherwise fill the queue and cause a burst of packet drops. In addition, Blue has been shown to have direct applicability to improving the performance of multimedia applications by reducing packet loss rates and queueing delays to networked applications such as interactive audio and video [2].

Since our proposed GREEN (*G*eneralized *R*andom *E*arly *E*vasion *N*etwork) router classifies and ensures fairness between flows, we compare GREEN with two flow-based AQM schemes that are derivatives of RED and Blue — Flow-based RED (FRED) [3] and Stochastic Fair Blue (SFB) [4]. While FRED and SFB *avoid* congestion by reacting to congestion early before it becomes too problematic, GREEN *prevents* congestion from ever occurring, thus providing more stable QoS over a best-effort network. This congestion-prevention PQM scheme is based on a mathematical model of the steady-state behavior of TCP [5] such that a high degree of fairness between flows is achieved while simultaneously maintaining high link utilization, low packet loss, and short queues. These characteristics are particularly beneficial to multimedia applications. Further, GREEN can also be used to identify and police non-TCP-friendly flows. However, we do not address this aspect of GREEN in this paper.

This work was supported by the U.S. Dept. of Energy's Laboratory-Directed Research & Development Program through Los Alamos National Laboratory contract W-7405-ENG-36. Apu Kapadia was funded in part by the U.S. Dept. of Energy's High-Performance Computer Science Fellowship through Los Alamos National Laboratory, Lawrence Livermore National Laboratory, and Sandia National Laboratory.

II. Algorithm

GREEN applies knowledge of the steady-state behavior of TCP connections at the router to intelligently drop (or mark) packets for congestion notification. By using such a mechanism, a router can give each connection its fair share of bandwidth while preventing the build-up of packet queues.

The throughput of a TCP connection depends, among other factors, on its round trip time (RTT) and the probability that its packets are dropped in the network. Specifically, Mathis et al. [5] show that a connection's throughput satisfies the following equation under certain simplifying assumptions:

$$BW = \frac{MSS \times c}{RTT \times \sqrt{p}} \qquad (1)$$

where BW is the bandwidth/throughput of the connection, MSS is the maximum segment size, RTT is its round trip time, and p is the packet loss probability. c is a constant that depends on the acknowledgement strategy that is used (e.g., delayed or every packet) as well as on whether packets are assumed to be lost periodically or at random.

While this model may not be applicable to non-SACK TCP implementations, to very short connections that never reach steady state, or to connections whose window size are artificially limited by the receiver's flow-control window. These issues are generally not problems in properly configured long-lived connections such as in multimedia applications, e.g., video-on-demand (VoD) and bulk data transfers.

Now, let us consider a scenario where there are N active flows at a router on a particular outgoing link of capacity L. GREEN considers a flow to be active if it has had at least one packet go through the router within a certain *window* of time. The fair-share throughput of each flow is L/N (assuming each source attempts to transmit at least at that rate). Substituting L/N for BW in Equation (1), we derive the following expression for loss probability p, i.e.,

$$p = \left(\frac{N \times MSS \times c}{L \times RTT} \right)^2 \qquad (2)$$

By using this value of p as the dropping probability for congestion notification, GREEN "coerces" flows into sending at their fair rate. Because p depends on the number of flows and the round-trip time of each flow, congestion notification is more aggressive for large N and small RTT. And by including RTT as an inverse parameter in the equation, GREEN eliminates the bias of favoring TCP connections with smaller $RTTs$ with respect to throughput [6]. (Recall that TCP connections with smaller $RTTs$ can increase their window size faster due to the smaller RTT and are therefore more aggressive. Hence, these flows grab more than their fair share of bandwidth, which leads to this bias.)

End-to-end schemes have been proposed to correct for this bias by requiring TCP senders to increase their congestion windows by a constant proportional to RTT^2 [7][8]. However these schemes rely on a *window constant*, which is hard to calculate and varies with the topology of the network. In contrast, not only can GREEN accurately calculate the drop probabilities irrespective of the network topology, it also does not require any end-to-end modifications. Hence, GREEN can be used at edge routers to ensure fairness between flows (without having to modify stock TCP implementations). For example, an organization could choose to implement GREEN at its edge router to ensure fairness between flows that begin or end within that organization.

The basic operation of GREEN does not require per-flow state information. N and MSS can be easily estimated, and in Section IV we discuss how the RTT for a flow can be inferred from the network. Hence GREEN has minimal state requirements. FRED keeps per-flow state information for flows that have packets buffered at the link. SFB does not maintain per-flow state information, but instead employs a Bloom filter [9] to hash flows into L levels of N bins. Each bin maintains queue occupancy statistics for flows that map into that bin and a corresponding drop probability p_m. Hence, the required state is $O(N * L)$. The selection of L and N is discussed in [4].

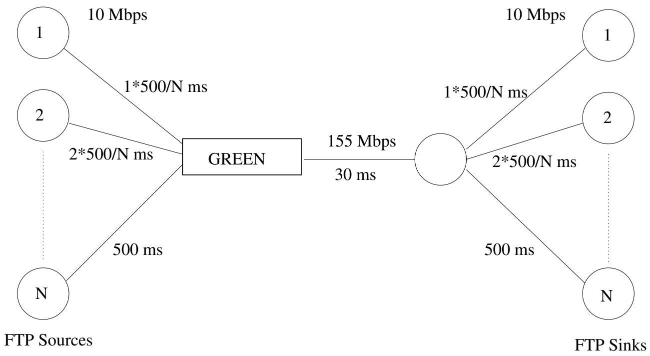


Fig. 1. Network Topology

III. Experiments and Evaluation

We evaluate the performance of GREEN with respect to Flow-based RED (FRED) [3] and Stochastic Fair Blue (SFB) [4]. We compare GREEN to FRED and SFB since these active queue-management schemes are also flow-based. We also include results for "Drop Tail" queueing to provide a baseline for assessing performance.

We assume that a router knows the bandwidth (L) of the attached outgoing link. N is the number of active flows, i.e., flows that have had at least one packet go through the router within a certain *window* of time. We discuss the estimation of N in Section IV.

The MSS of a flow is estimated by the router by looking at the size of each packet. By doing this, we reduce the need for flow state to maintain information regarding packet sizes seen over an interval. In our experiments, we chose MSS to be 1 KB in all cases. The value of c, in our "random dropping, delayed acknowledgment" model was fixed at 0.93 [5]. For the time being, we assume that the router has some mechanism for determining a flow's RTT and briefly discuss how this parameter can be estimated in Section IV.

We used *ns* [10] to evaluate the performance of GREEN over a network with the topology shown in Fig. 1. N sources and N sinks are connected to the routers over 10-Mbps links with propagation delays ranging from $\frac{500}{N}$ ms to 500 ms. We varied the number of flows, N from 50 to 500. The bottleneck link has a bandwidth of $155Mbps$ and a delay of 30 ms.[1]

For our simulations, we started FTP connections from the leftmost nodes to the rightmost nodes within the first second of simulation and ran it for 180 seconds. GREEN is implemented at the gateway, which is the bottleneck router in our simulation. All of the metrics presented in this section — link utilization, fairness, packet loss, queue size — are measured at this gateway. We present results for link utilization, fairness, and queue size, after the first 50 seconds to remove the startup transient effects and to study the steady-state behavior.

[1] Since we simulate more than 50 flows over $10Mbps$ links, the $155Mbps$ link creates a bottleneck.

A. Fairness

GREEN attempts to limit all the TCP flows to their fair share of the outgoing link bandwidth. To assess GREEN's ability to maintain equal bandwidths between TCP flows, we use Jain's Fairness Index [11]. We briefly describe how the fairness index is calculated and then present our results.

1) Jain's Fairness Index: This fairness index quantifies the degree of similarity between all the flow bandwidths. Given the set of throughputs $(x_1, x_2, \ldots, x_n)$, the fairness index is calculated as follows:

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

The fairness index always lies between 0 and 1. Hence, a higher fairness index indicates better fairness between flows. The fairness index is 1 when all the throughputs are equal. When all the throughputs are not equal, the fairness index drops below 1.

2) Results: As shown in Fig. 2, GREEN provides significantly better bandwidth fairness than the other queue-management schemes. The curve for Drop Tail shows us the fairness that would be expected at most gateways in the Internet today. FRED outperforms Drop Tail and SFB because it queues at least two packets[2] of a flow before marking a packet from that flow. This provides much better fairness as long as each flow maintains one to two outstanding packets at the gateway. SFB exhibits poor fairness because it is sensitive to varying RTTs between flows and breaks down under a large number of connections with varying RTTs [4].

GREEN's fairness index gradually rises. As the number of flows increases, we observe a lower fairness index for fewer flows because each flow has a higher share of bandwidth and flows with longer RTTs are unable to attain their steady state bandwidths in 180 seconds. Why? Even though GREEN can "slow down" flows with shorter RTTs (by dropping packets), it cannot speed up flows with longer RTTs. Better fairness for fewer flows is achieved by increasing the simulation time, allowing all flows to reach their steady state bandwidths. With a larger number of flows, the fair share of bandwidth is low enough so that all flows are able to attain close to their average share of bandwidth. Longer simulations yield even better fairness curves for Green while showing little improvement for Drop Tail, FRED, and SFB.

B. Link Utilization

At the end of each simulation, the overall link utilization is calculated as follows:

$$utilization = \frac{total\ byte\ departures\ in\ t\ sec}{bandwidth \times t}$$

The numerator equals the total number of bytes leaving the link during the interval of t seconds, and the denominator equals

[2] In our experiments, we operate FRED under the *"many flow"* mode.

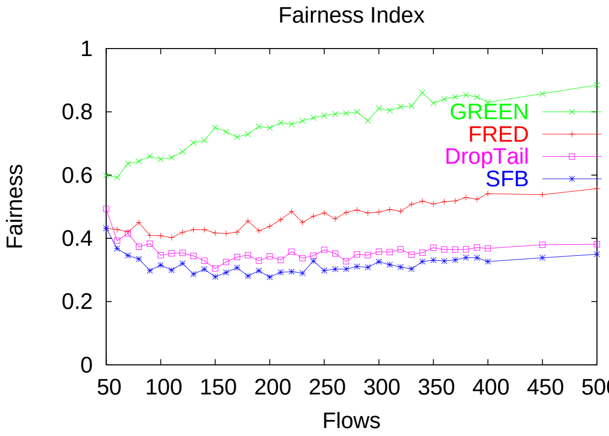


Fig. 2. Jain's Fairness Index vs. Number of Flows

the total possible bytes that could have left the link in the same interval.

Fig. 3(a) shows that as the number of flows increases, GREEN performs as well as SFB in terms of link utilization. Drop Tail achieves higher utilizations because the flows with shorter RTTs are allowed to be aggressive and no packets are voluntarily dropped early. While this yields better link utilizations, it sacrifices fairness heavily and can lead to higher packet loss. GREEN outperforms FRED with respect to fairness *and* link utilization because GREEN deterministically maintains the average bandwidths of all flows while FRED simply relies on queue occupancy statistics to regulate queue size.

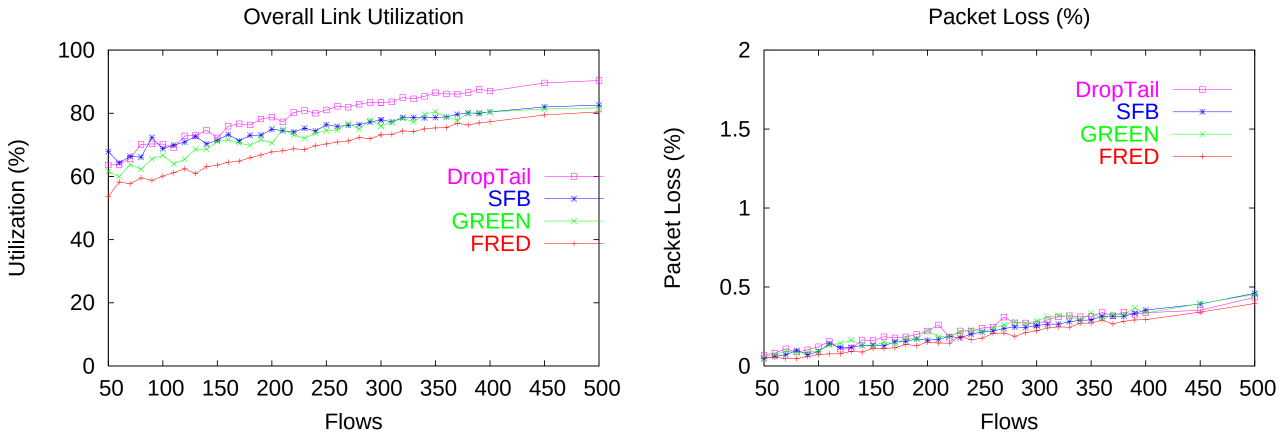
C. Packet Loss

As shown in Fig. 3(b), the packet loss percentage is roughly the same for all flows and stays below 0.5%. Equation (1) provides good estimates for $p < 1\%$ [5]. Since the overall packet loss stays well below 1% in our simulations, GREEN limits the rates of flows to their equal shares of bandwidth effectively.

D. Queue Size

Fig. 4(a) shows the queue sizes for Drop Tail for 100 flows and 500 flows. As we can see, as the number of flows increases, the average queue size for Drop Tail increases dramatically. In contrast, as seen in Fig. 4(b), 4(c) and 4(d), GREEN, FRED and SFB keep the average queue sizes low.

FRED keeps queue sizes low by marking packets beyond a certain threshold and limiting the amount of buffer space for each flow. SFB does so by increasing drop rates when there is packet loss and reducing drop rates when the link is underutilized. Hence, FRED and SFB try to dynamically converge to the correct "operating point" for low queue sizes. GREEN achieves its operating point by calculating drop probabilities for each flow based on their fair share of bandwidth. By ensuring that the aggregate bandwidth of the flows is equal to the available bandwidth at the link, there is no sustained buildup in queue length.



(a) Overall Link Utilization vs. Number of Flows

(b) Overall Packet Loss vs. Number of Flows

Fig. 3. Link Utilization and Packet Loss statistics

IV. Estimating Flow Parameters

Up until now, we assumed that routers had knowledge of the various flow parameters: MSS, N, and RTT. The parameters MSS and N can be easily estimated. The MSS of a flow is estimated by the router by looking at the size of each packet. One way that N can be estimated is by counting the number of flows that have had at least one packet go through the router within a certain *window* of time. Long windows may cause GREEN to overestimate the number of flows and reduce the overall link utilization. Short windows may cause GREEN to underestimate the number of flows, resulting in over-provisioning of the link bandwidth. In our experiments, we assumed FTP file transfers, and hence the number of flows was a constant. Stabilized RED (SRED) uses a statistical technique to estimate the number of active flows [12]. SRED compares an arriving packet with a randomly chosen packet that was seen recently. If these packets belong to the same flow, the authors call it a *hit*. Hit rates can be statistically analyzed to give a reasonable estimate of the number of active flows, N. Hit rates can also be used to identify and limit non-responsive flows that use more than their fair share of bandwidth.

The round trip time, RTT, is the most difficult parameter to estimate with any degree of accuracy. Below we briefly present two possible solutions to this. One solution makes use of the fact that the best estimates of RTT occurs at the TCP end host. Using the IP options available, the sender can insert the RTT into a prespecified field in the IP header. This can then be utilized by routers along the way where GREEN is deployed. Another solution is to make use of the IDMaps service [13]. A GREEN router can also perform the duties of an IDMaps *tracer* and maintain a database of RTT estimates, which can be used for fast lookups based on source and destination IP addresses. IDMaps provides RTT estimates accurate to within twice the actual value. Such estimates could result in lower fairness but are sufficient for GREEN to still outperform FRED, SFB and Drop Tail in terms of fairness. We leave further discussion and analysis of the use of IDMaps for future work.

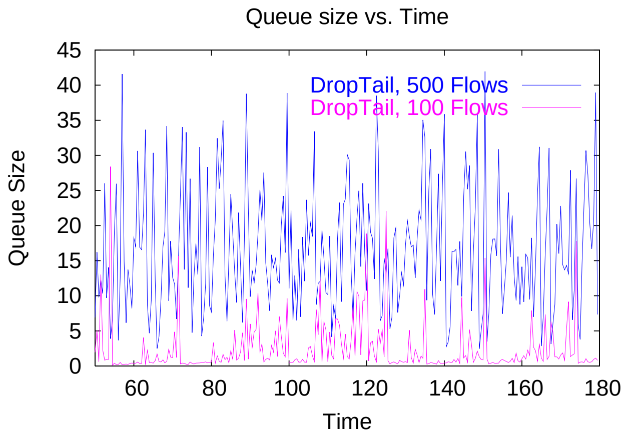
V. Conclusion

In this paper, we have shown how applying the knowledge of the macroscopic behavior of the TCP congestion-avoidance algorithm at the routers can be used to achieve a high degree of fairness between flows while simultaneously maintaining high link utilization, low packet loss, and short queues. And because the implementation of GREEN relies on the knowledge of flow RTTs and the total number of active flows, we briefly discussed how these metrics can be inferred by a GREEN router.

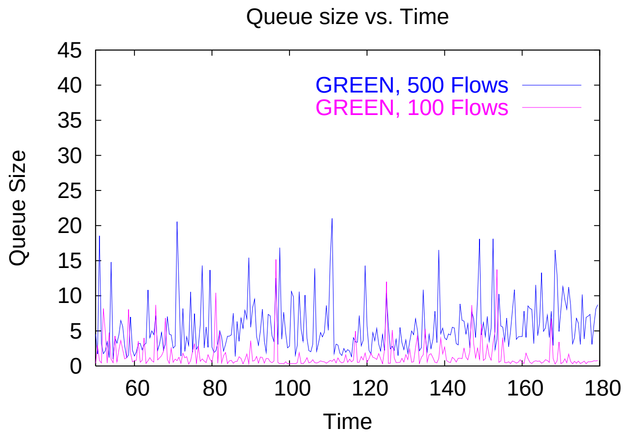
Similar to FRED and SFB, GREEN requires very little state information for its operation. Future work will focus on (1) identifying and policing non-responsive flows such as misbehaving real-time multimedia traffic and denial of service attacks, (2) adding a Bloom filter (as in SFB) or a hit-rate module (as in SRED) to identify and limit unresponsive flows, and (3) examining how GREEN behaves with short-lived connections (e.g., so-called ”web mice”) or with mixtures of different traffic types. The applications that stand to benefit the most from the “fixed low-latency/high throughput fairness property” of this algorithm are delay-sensitive traffic such as streaming, real-time audio and video applications.

References

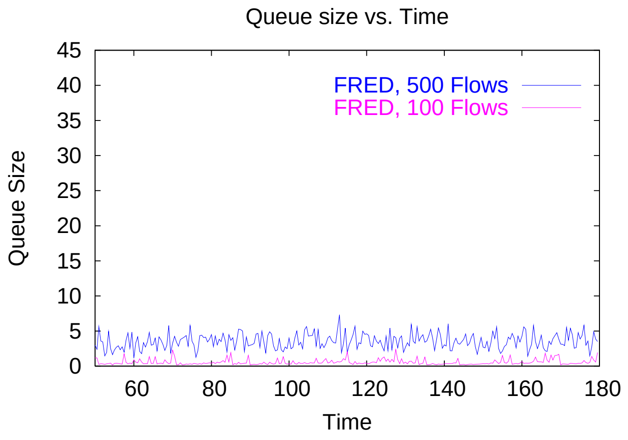
[1] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.
[2] W. Feng, D. Kandlur, D. Saha, and K. Shin, “BLUE: A New Class of Active Queue Management Algorithms,” Tech. Rep., April 1999.
[3] D. Lin and R. Morris, “Dynamics of Random Early Detection,” in *Proc. of ACM SIGCOMM 1997*, September 1997.



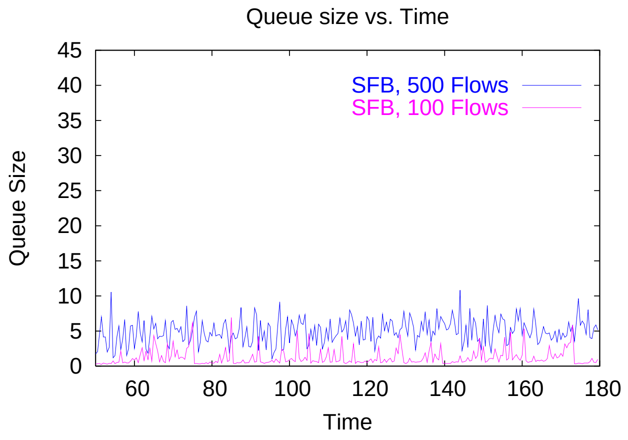
(a) Drop Tail: Queue Size vs. Time



(b) GREEN: Queue Size vs. Time



(c) FRED: Queue Size vs. Time



(d) SFB: Queue Size vs. Time

Fig. 4. Queue Occupancy Statistics for Drop Tail, Green, FRED, and SFB

[4] W. Feng, D. Kandlur, D. Saha, and K. Shin, "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness," in *Proc. of IEEE INFOCOM 2001*, April 2001.

[5] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *Computer Communication Review*, vol. 27, no. 3, July 1997.

[6] T. Lakshman and U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, June 1997.

[7] S. Floyd, "Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic," *ACM Computer Communication Review*, vol. 21, no. 5, pp. 30–47, Oct. 1991.

[8] T. Henderson, E. Sahouria, S. McCanne, and R. Katz, "On Improving the Fairness of TCP Congestion Avoidance," in *Proc. of IEEE Globecom*, November 1998.

[9] B. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM*, vol. 13, no. 7, July 1970.

[10] ns, "Network Simulator," http://www.isi.edu/nsnam/ns.

[11] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, 1989.

[12] Teunis J. Ott, T. V. Lakshman, and Larry H. Wong, "SRED: Stabilized RED," in *Proceedings of INFOCOM*, 1999, vol. 3, pp. 1346–1355.

[13] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: A Global Internet Host Distance Estimation Service," *IEEE/ACM Transactions on Networking*, 2001 October.