LA-UR- 02-6126

Approved for public release; distribution is unlimited.

Title: GREEN + IDMaps: A Practical Solution for Ensuring Fairness in a Biased Internet

Author(s): A. Kapadia, S. Thulasidasan, and W. Feng

Submitted to: IEEE International Conference on Communications

Los Alamos
NATIONAL LABORATORY

GREEN + IDMaps: A Practical Solution for Ensuring Fairness in a Biased Internet

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

[†] Computer & Computational Sciences Research 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—**GREEN is a proactive queue-management (PQM) algorithm that removes TCP's bias against connections with longer round-trip times, while maintaining high link utilization and low packet-loss. GREEN applies knowledge of the steady-state behavior of TCP connections to proactively drop packets, thus preventing congestion from ever occurring. As a result, GREEN ensures much higher fairness between flows than other active queue management schemes like Flow Random Early Drop (FRED) and Stochastic Fair Blue (SFB), which suffer in topologies where a large number of flows have widely varying round-trip times.**

GREEN's performance relies on its ability to gauge a flow's round-trip time (RTT). In previous work, we presented results for an ideal GREEN router which has accurate RTT information for a flow. In this paper, we present a practical solution based on IDMaps, an Internet distance-estimation service, and compare its performance to an ideal GREEN router. We show that a solution based on IDMaps is practical and maintains high fairness and link utilization, and low packet-loss rates.

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 [7] and Blue [3] 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. Flow Random Early Drop (FRED) [11] and Stochastic Fair Blue (SFB) [4] improve on the performance of RED and BLUE and operate at the flow level.

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 is funded 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 Laboratories.

While FRED and SFB *avoid* congestion by reacting to it before it becomes problematic, our *proactive* queue-management (PQM) algorithm, called GREEN [5], *prevents* congestion from ever occurring. This congestion-prevention PQM scheme is based on a mathematical model of the steady-state behavior of TCP [12] such that flows can be regulated to receive their fair share of the bottleneck link bandwidth while simultaneously maintaining high link utilization and low packet-loss. However, this mathematical model relies on the ability of the router to infer the round-trip time (RTT) of a flow. In [5] we presented results for an *ideal* GREEN (we will refer to this as *GREEN-Ideal*) router that was assumed to have access to this information. In this paper we present a practical solution for GREEN based on IDMaps [8], an Internet distance estimation service. While IDMaps does not provide exact RTT estimates, we examine the impact of an IDMaps-based solution for GREEN and compare it with GREEN-Ideal. We show that even in the face of slightly inaccurate estimates, provided by IDMaps, GREEN still outperforms other flow-based AQM schemes like FRED and SFB, while still maintaining high link utilization and low packet-loss.

II. Algorithm

Our PQM algorithm called 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. 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. [12] 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 acknowledgment strategy that is used (i.e., delayed or every packet) as well as on whether packets are assumed to be lost periodically or at random.

In general, this model may not be applicable to non-SACK TCP implementations in environments where there are sustained multiple packet-losses for a flow within a single RTT (causing repeated timeouts). This model may also not apply to very short connections that never reach steady state, or to connections whose window sizes are artificially limited by the receiver's flow control window. We assume that all connections satisfy the assumptions required for this model.

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 1 packet go through the router within a certain *window* of time. For now we assume that this parameter can be easily estimated, and briefly explain this parameter in Section VI. 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 (1), we derive the following expression for loss probability p:

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

By using this value of p as the dropping probability for congestion notification, GREEN "coerces" flows into sending at their fair rate. Note that GREEN applies this marking probability to *all* arriving packets, where the value of p depends on the flow. 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 [10]. (Recall that TCP connections with smaller $RTTs$ can increase their window size faster due to the smaller RTT, and are more aggressive. These flows are able to grab more than their fair share of bandwidth, which leads to this bias.)

III. Placement of GREEN Routers

GREEN is mainly suited as an edge router, where organizations can enforce fairness between flows leaving the organization through a bottleneck link. In a situation with widely varying RTTs, it is desirable to correct TCP's bias and ensure fairness between flows while maintaining high link utilization and low packet-loss. In contrast, 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 [6][15]. However, these schemes rely on a *window constant* that 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 network topology, it also does not require any end-to-end modifications.

IV. RTT Estimation using IDMaps

In [5], we presented results for GREEN-Ideal, where the RTT was assumed to be known at the router. Here, we relax this constraint by making use of IDMaps. IDMaps [8] is a scalable Internet-wide service that aims to provide Internet distance estimates. For example, the authors have suggested that IDMaps can be used by hosts for nearest mirror selection. Such a service is also well suited to GREEN, which can obtain RTT estimates for flows using IDMaps. We propose an architecture where GREEN routers are part of the IDMaps framework, and therefore, can perform fast lookups in a local IDMaps database.

A. IDMaps - Architecture

Jamin et al. [8] argue that providing highly accurate delay estimates (within 5% for example) is not feasible. Instead they aim to provide a scalable solution with existing technology to provide delay estimates that are accurate to within a factor of two. Jamin et al. propose the deployment of *tracers* in the Internet. Tracers maintain raw distances amongst themselves and address prefixes (AP). The use of APs, as opposed to actual IP addresses, makes this solution feasible, trading off accuracy for scalability. The delay between two IP addresses is estimated by calculating the sum of the delays between the two tracers closest to the two address prefixes, and the tracer-AP delays. The IDMaps Project [9] is already running an experimental service and can be accessed at *http://www.closestserver.com/*.

B. GREEN using IDMaps

We propose a solution in which GREEN routers also perform the duties of tracers and exchange distance information with other tracers. We do not expect this to add much overhead to existing traffic from routing updates. Furthermore, since GREEN is an edge router, the delays from sources within the organization to the GREEN router will be fairly low. GREEN can perform fast lookups in the local IDMaps database to obtain RTT estimates for a flow based on the destination IP addresses (since the source IP address is assumed to be within the organization). GREEN calculates the drop probability based on the *estimated RTT*. The accuracy of IDMaps estimates is sensitive to the number of tracers and their placement on the Internet. Jamin et al. have evaluated several graph-theoretic approaches as well as simple heuristics. In general, the accuracy of estimates increases when tracers are closer to the APs. As mentioned earlier, GREEN routers will be co-located with the APs of that organization, and hence, will result in more accurate estimates.

V. State Requirements

The basic operation of GREEN does not require per-flow state information. N and MSS can be easily estimated. Since we propose that GREEN routers operate as IDMaps tracers, GREEN will maintain state proportional to the number of tracers deployed in the Internet. The amount of state used depends on how the tracers are connected through virtual links. This is discussed in more detail in [8]. 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 [1] 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, SFB's state requirement is $O(N * L)$. A discussion on the selection of L and N is discussed in [4].

VI. Experiments and Evaluation

Here, we evaluate the performance of GREEN with IDMaps (GREEN-IDMaps) with GREEN-Ideal. We also compare the performance with respect to FRED and SFB. We make comparisons with 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 1 packet go through the router within a certain *window* of time. We discuss the estimation of N in Section VII. The MSS of a flow is estimated by the router by looking at the size of each packet. 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 [12].

Our simulations for GREEN-Ideal are based on the actual RTTs and show the best-case performance of GREEN. Since IDMaps aims to provide delay estimates within a factor of two, we modify our original simulations to simulate the effect of IDMaps. For GREEN-IDMaps, we set the RTT estimate to be a uniformly random number between one and two times the actual RTT. This will give us an idea of the worst-case performance of GREEN-IDMaps, assuming that IDMaps will usually provide better estimates than this.

We used *ns* [13] to evaluate the performance of GREEN over a network with the topology shown in Fig. 1. We try to simulate an organizational topology with low latencies to the "left" of the bottleneck edge router (GREEN). We simulate connections of varying RTTs on the "right" and vary their latencies uniformly from $1ms$ to $500ms$. N Sources and N sinks are connected to the routers over $10Mbps$ links. We varied the number of flows N, from 50 to 500. The bottleneck link has a bandwidth of 155 Mbps and a delay of 30 ms.

We started FTP connections from the leftmost nodes to the rightmost nodes within the first second of simulation and ran it for 420 seconds. GREEN-Ideal and GREEN-IDMaps were 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.

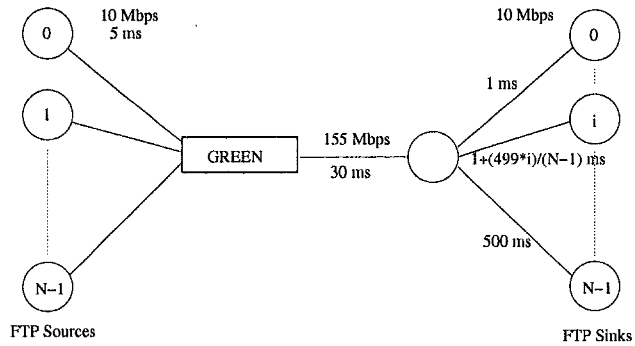


Fig. 1. Network Topology

A. Fairness

As mentioned in Section II, GREEN attempts to regulate all the TCP flows to their fair share of the outgoing link bandwidth. We use Jain's Fairness Index [2] to assess GREEN's ability to maintain equal bandwidths between TCP flows. We briefly describe how the fairness index is calculated, and then present our results.

1) Jain's Fairness Index: 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-Ideal provides significantly higher bandwidth fairness than the other queue management schemes. The curve for Drop Tail shows us the fairness we would expect at most gateways in the Internet today. FRED is able to outperform Drop Tail and SFB because it queues at least two packets[1] 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]. Most importantly, we can see that GREEN-IDMaps is able to achieve fairness significantly better than FRED, SFB, and Drop Tail despite the

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

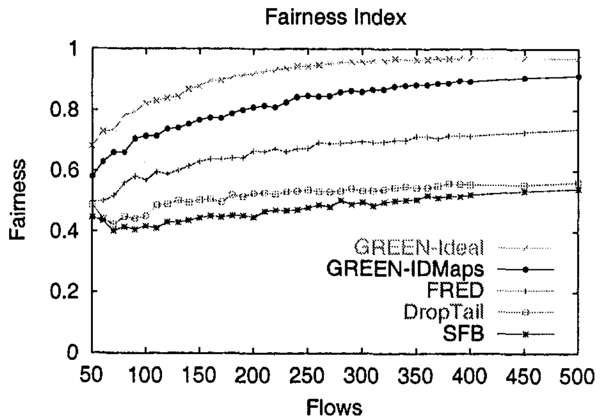
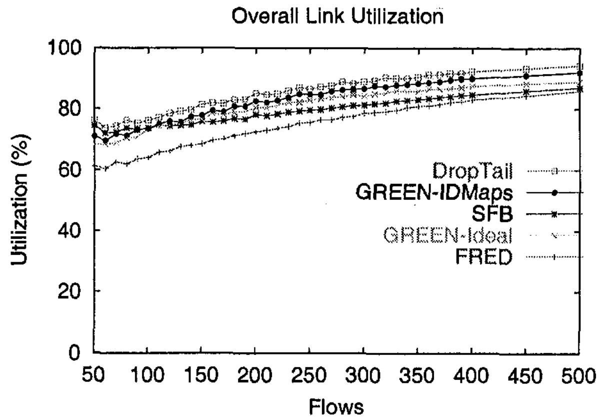
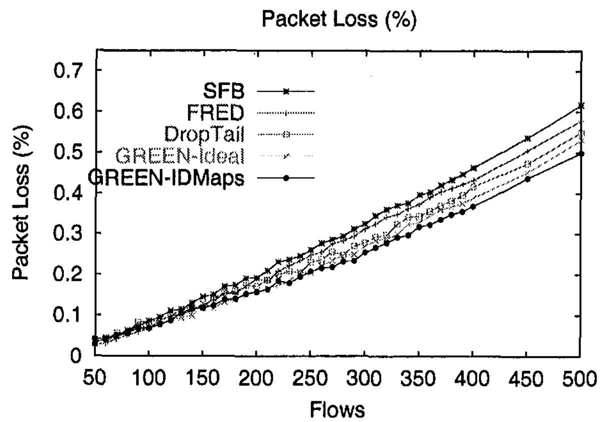


Fig. 2. Jain's Fairness Index vs. Number of Flows



(a) Overall Link Utilization vs. Number of Flows



(b) Overall Packet-Loss vs. Number of Flows

Fig. 3. Link Utilization and Packet-Loss statistics

rough RTT estimates. While the fairness provided by GREEN-IDMaps is not as good as that provided by GREEN-Ideal, we can see that a solution based on IDMaps is indeed practical and can be deployed in the Internet.

We also see that GREEN's fairness index gradually rises with the number of flows. 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 420 seconds. 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.

B. Link Utilization

Here we compare GREEN's performance with SFB, FRED and Drop Tail in terms of overall 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 the total possible bytes that could have left the link in the same interval.

Fig. 3(a) shows that GREEN-Ideal achieves higher link utilization than SFB and FRED because GREEN can deterministically maintain the average bandwidths of all flows, while FRED simply relies on queue-occupancy statistics to regulate queue size, and SFB relies on link utilization and packet-loss statistics to regulate queue size. Drop Tail achieves higher utilization because the flows with shorter RTTs are allowed to be aggressive. While this yields better link utilizations, it sacrifices fairness heavily, as seen in Fig. 2. In all cases, GREEN-IDMaps achieves better utilization than GREEN-Ideal because GREEN-IDMaps overestimates the RTTs, which results in lower dropping probabilities. This in turn results in over-subscribing of the available bandwidth, which results in higher sending rates for all the flows and more queueing at the GREEN-IDMaps router. However, as noted in Section VI-A, GREEN-IDMaps provides superior fairness compared to Drop Tail, FRED, and SFB. This makes the version of GREEN based on IDMaps attractive since it has high link utilization as well as a high fairness index.

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 $p < 1\%$ [12]. Since the overall packet-loss stays well below 1% in our simulations, both GREEN-Ideal and GREEN-IDMaps are able to limit the rates of flows to their fair share of bandwidth. As mentioned in Section VI-B, GREEN-IDMaps underestimates the drop probabilities because it overestimates the RTTs. This is why we observe that GREEN-IDMaps has packet-loss rates lower than GREEN-Ideal.

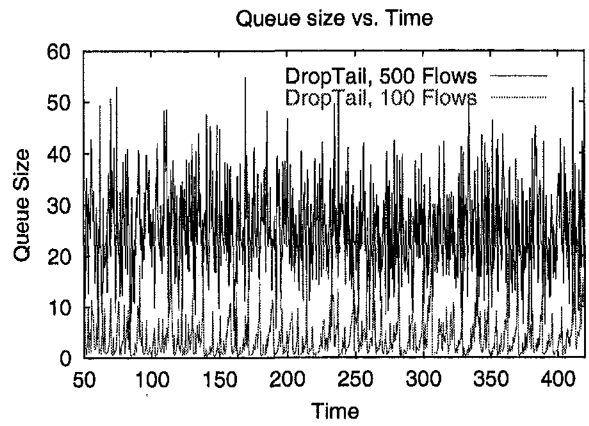
D. *Queue Size*

Fig. 4(a) shows the queue sizes for Drop Tail for 100 flows and 500 flows. As the number of flows increases from 100 to 500, the average queue size for Drop Tail increases dramatically. In contrast, as seen in Fig. 4(b), 5(a), and 5(b), GREEN-Ideal, FRED, and SFB are able to keep the average queue sizes low. GREEN-IDMaps' performance (Fig. 4(c)) lies between that of GREEN-Ideal and Drop Tail. Even though the increase in queue lengths is not as dramatic as in Drop Tail, we can see how the rough RTT estimation affects GREEN-IDMaps.

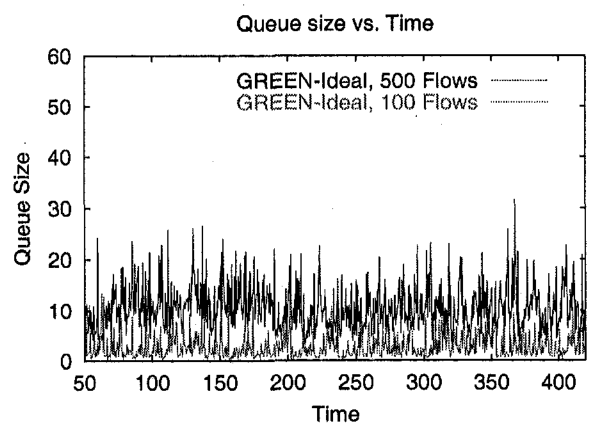
FRED keeps queue sizes low by marking packets beyond a certain threshold and limiting the amount of buffer-space for each flow. SFB does so by increasing drop rates when there is packet-loss and reducing drop rates when the link is underutilized. Hence, FRED and SFB attempt to dynamically converge to the correct "operating point" for low queue sizes. GREEN-Ideal 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. GREEN-IDMaps over-subscribes the available bandwidth, which results in queues building up and exhibiting Drop Tail-like behavior when the link capacity is reached.

VII. Estimating Flow Parameters

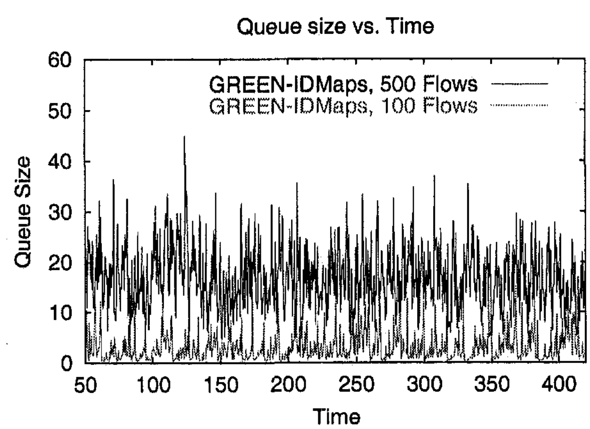
GREEN relies on its ability to estimate N, MSS and RTT. N and MSS can be easily estimated. The MSS of a flow is estimated by the router by looking at the size of each packet. One way that N can be estimated is by counting the number of flows that have had at least one packet go through the router within a certain *window* of time. Longer windows may cause GREEN to overestimate the number of flows and reduce the overall link utilization. Shorter windows may cause GREEN to underestimate the number of flows, resulting in over-subscribing 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 [14]. 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. We have already discussed RTT estimation using IDMaps.



(a) Drop Tail: Queue Size vs. Time

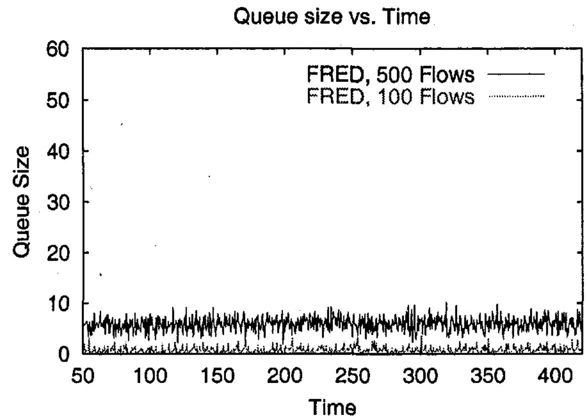


(b) GREEN-Ideal: Queue Size vs. Time

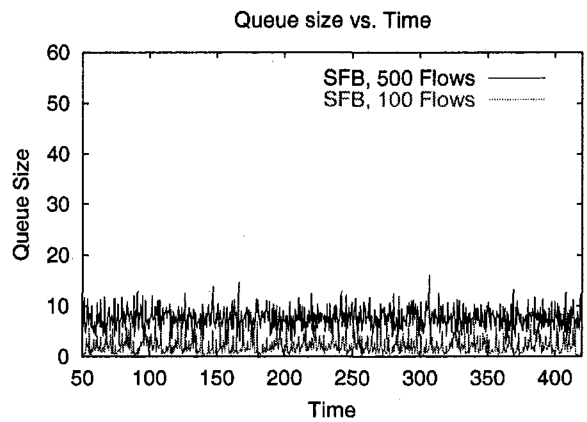


(c) GREEN-IDMaps: Queue Size vs. Time

Fig. 4. Queue Occupancy Statistics for Drop Tail, GREEN-Ideal, and GREEN-IDMaps



(a) FRED: Queue Size vs. Time



(b) SFB: Queue Size vs. Time

Fig. 5. Queue Occupancy Statistics for FRED and SFB

VIII. Conclusion

In [5], we showed how GREEN performed under ideal circumstances. GREEN corrects TCP's bias in situations with a large number of flows with varying RTTs while ensuring high fairness. In this paper, we explored a practical implementation of GREEN based on IDMaps. We simulated a realistic topology and showed how GREEN would perform as an edge router, providing fairness and high link utilization for an organizational network. The main tradeoff, as a result of using IDMaps to estimate RTTs, was that GREEN-IDMaps exhibited slightly lower fairness than GREEN-Ideal, while still outperforming FRED, SFB, and Drop Tail. And as a consequence of overestimating RTTs for flows, GREEN-IDMaps had higher utilizations than GREEN-Ideal, lower packet-losses, and larger queue lengths.

Future investigations will include how this mechanism behaves for short-lived connections and mixtures of different traffic types. To deal with TCP-unfriendly traffic, we also plan on integrating functionality (e.g., like that in SFB and SRED) to identify and limit unresponsive flows that use more than their fair share of bandwidth.

References

[1] B. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM*, 13(7), July 1970.
[2] D. Chiu and R. Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.
[3] W. Feng, D. Kandlur, D. Saha, and K. Shin. BLUE: A New Class of Active Queue Management Algorithms. Technical report, April 1999.
[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] W. Feng, A. Kapadia, and S. Thulasidasan. GREEN: Proactive Queue Management over a Best-Effort Network. In *Proc. of IEEE Globecom 2002*, November 2002.
[6] S. Floyd. Connections with multiple congested gateways in packet-switched networks part1: One-way traffic. *ACM Computer Communication Review*, 21(5):30–47, October 1991.
[7] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
[8] 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.
[9] Internet Distance Maps Service. http://idmaps.eecs.umich.edu/.
[10] 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*, 5(3):336–350, June 1997.
[11] D. Lin and R. Morris. Dynamics of Random Early Detection. In *Proc. of ACM SIGCOMM 1997*, September 1997.
[12] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *Computer Communication Review*, 27(3), July 1997.
[13] ns. Network Simulator. http://www.isi.edu/nsnam/ns.
[14] T. J. Ott, T. V. Lakshman, and L. H. Wong. SRED: Stabilized RED. In *Proc. of IEEE INFOCOM 1999*, March 1999.
[15] T.Henderson, E. Sahouria, S. McCanne, and R. Katz. On Improving the Fairness of TCP Congestion Avoidance. In *Proc. of IEEE Globecom*, November 1998.