ON STANDARDIZED NETWORK TOPOLOGIES FOR NETWORK RESEARCH

George F. Riley

Department of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250

ABSTRACT

Simulation has become the evaluation method of choice for many areas of computer networking research. When designing new or revised transport protocols, queuing methods, routing protocols, (just to name a few), a common approach is to create a simulation of a small to moderate scale topology and measure the performance of the new methodology as compared to existing methods. We demonstrate that simulation results using this approach can lead to very misleading, and even incorrect, results. The interaction between the large number of variables in these simulations can lead to results that vary widely from between different simulation topologies. We give empirical evidence showing different conclusions when the same comparisons are done using differing topologies. We argue the need for a standardized taxonomy of simulation topologies that capture a significant and realistic range of values for the various variables that impact the performance of a simulated network.

1 INTRODUCTION

Does the Random Early Detection (*RED*) (Floyd and Jacobson 1993) queuing method give better web browsing response than the simpler First–In–First–Out (*FIFO*) method (Christiansen et al. 2000)? Will the TCP Selective–Acknowledgment (*SACK*) feature (Fall and Floyd 1996; Bruyeron, Hemon, and Zhang 1998) improve the performance of bulk data transfer on the Internet? Can greedy Internet users get better TCP performance by disabling congestion control (Jacobson 1988)? What is the overall affect of random packet reordering (Bennett, Partridge, and Schectman 1999) on long–lived Internet TCP flows? These are but a few of the interesting and important questions that lead to use of simulation methods to seek answers. Unfortunately, the answer to these and many similar questions is "it depends".

The behavior and performance of a complex protocol such as TCP, when deployed on a complex network, can be highly variable. Even very small changes in initial conditions or parameter settings can lead to large variations in performance metrics (as we later demonstrate). The interactions between queue length, round trip time, congestion windows, receiver windows, re-transmit timers, round trip estimations, and lost packet re-transmissions (just to name a few) are complex and lead to performance metrics that vary as these parameters interact.

We propose the standardization of a large set of simulation topologies that captures a complete set of reasonable values for these interacting variables. Any simulation based comparative study would use every one of the standardized simulation topologies and collect aggregate performance metrics for the standardized set, taken as a whole. We doubt that any comparative analysis would show a clear winner in every single topology instance, but expect that taken as an aggregate of all the topologies, clear and correct conclusions could be reached.

The remainder of this paper is organized as follows. Section 2 gives some representative case studies that demonstrate the need for the standardized topology set. Section 3 outlines a taxonomy of the variables and range of reasonable values that would be used to construct the standardized topology set. Section 4 gives some conclusions and future directions of this work.

2 SIMULATION CASE STUDIES

In this section, we give two example simulation studies that have led to conflicting results, depending on changes in the simulated topology or parameters settings.

2.1 TCP Cheaters

In this simulation study, we set out to determine if *rouge* or *cheater* TCP implementations (those that intentionally disable the TCP slow–start and congestion control mechanisms) could gain an unfair advantage over *normal* flows. The simulation topology was a variation on the network dis-
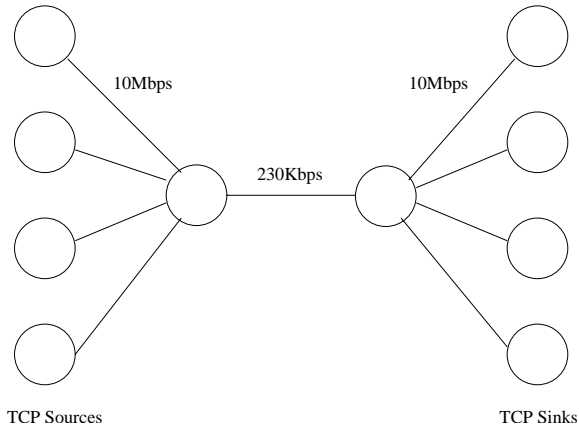
Figure 1: Cheater TCP Topology

cussed in (Jacobson 1988), and is shown in Figure 1. We used the *ns2* network simulator (McCanne and Floyd 1997), and created a version of TCP that uses neither slow-start nor congestion window management, called *TCP/NCC*. We then measured the performance of *TCP/NCC* as compared to normal TCP flows. The experiments use four competing flows, each starting three seconds apart. For each set of experiments, one simulation measured four normal TCP flows competing with each other, and a second simulation measured a single cheater flow competing with three normal flows. In both cases, each flow generated 2,048 packets. A total of three sets of experiments were performed, each using slight variations in the simulation parameters.

The results from the first set of experiments, shown in Figures 2 and 3 indicate virtually no difference at all in performance between the cheaters and normal TCP flows. In Figure 3, the cheater flow finishes at approximately time 160, virtually identical to the corresponding finish time for a normal flow shown in Figure 2. From this set of experiments we would conclude that a cheater TCP has little or no advantage over normal TCP.

The maximum queue length at the bottleneck link buffer was changed, and the experiments repeated. The results from this set of experiments is shown in Figures 4 and 5. In this case, the cheater flow appears to have a clear performance advantage. In Figure 5, the cheater flow finishes at time 100, well ahead of the corresponding normal flow in Figure 4. It's also interesting to note in this set of experiments that the normal TCP flow that starts first performs substantially better than the others in both cases. From this set of experiments we would conclude that a cheater TCP gains a clear advantage over normal TCP.

The maximum queue length at the bottleneck link buffer was changed again, and the experiments repeated. The results from this set of experiments is shown in Figures 6 and 7. This set of experiments shows clearly that the cheater TCP performs very poorly, not completing the transfer until

time 400, while the normal TCP flows all finish at about time 150 to 180.

Does a cheater TCP without congestion control have a performance advantage over normal TCP flows? The answer is clearly "it depends." We point out that this set of experiments is a contrived example, created to illustrate this point. Any researcher familiar with TCP congestion control mechanisms, and their interaction with bottleneck buffer limits, could deduce the behavior demonstrated here simply by thinking about it.

2.2 TCP Variations and Red vs. DropTail

In this section, we present a more realistic set of simulation experiments that again demonstrate the variation in comparative results obtained on differing topologies. In a recent graduate level class on network simulation, we asked the students to design and implement a simulation based study of the effectiveness of the Random Early Detection (*RED*) (Floyd and Jacobson 1993) queuing discipline, as compared to the simpler and more prevalent *DropTail* (*FIFO*) method. Furthermore, students were to compare four different TCP variations (Tahoe, Reno (Bruyeron, Hemon, and Zhang 1998), NewReno (Floyd and Henderson 1999), and SACK (Mathis et al. 1996)), and present results showing which of the four variations was "best". The students were given no instructions regarding the topology to use for the study, excepting that the topology for a similar experiment by Jeffay (Christiansen et al. 2000) was "too simple". Additionally, they were asked to perform this study using a "variety of network conditions", which was intentionally vague to give the students opportunity to decide on their own what those conditions should be. They were given a rough idea of the type of traffic to be modeled (loosely based on web browsing activity), but were again free to choose differing traffic conditions as warranted by their study.

The students were divided into eleven teams of about four students each, and were given three weeks to design, implement, and perform the study. Clearly, the topologies determined by each of the groups would be different in a number of variables, including number of bottlenecks, bottleneck bandwidth, bottleneck queue length, average round trip times, to name a few. A representative topology chosen by one group is shown in Figure 8.

The conclusions reached by the groups are quite interesting. Table 1 shows an exact 50/50 split, with four groups concluding that *RED* is better, four groups concluding *DropTail* is better, and three groups concluding there is no appreciable difference. Table 2 shows that every single group concluded the Selective Acknowledgment feature of TCP produced the best results.
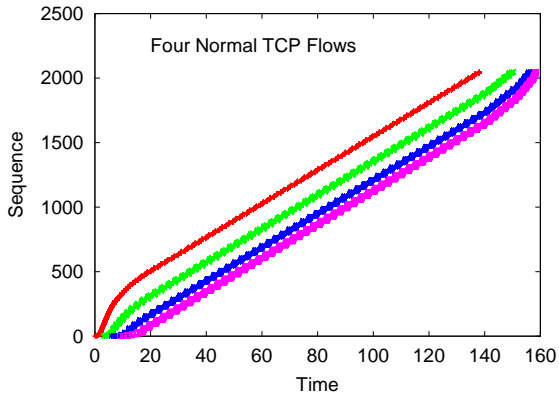
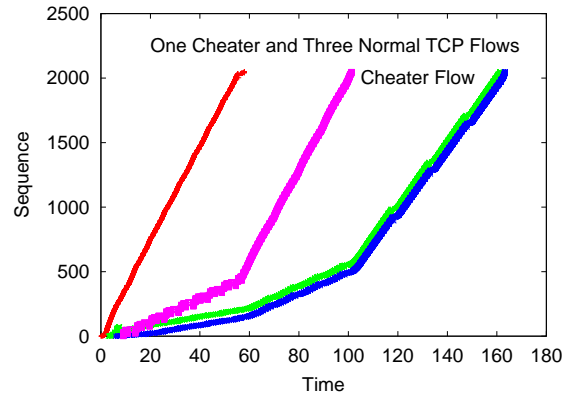Figure 2: Experiment 1, Four Normal TCP

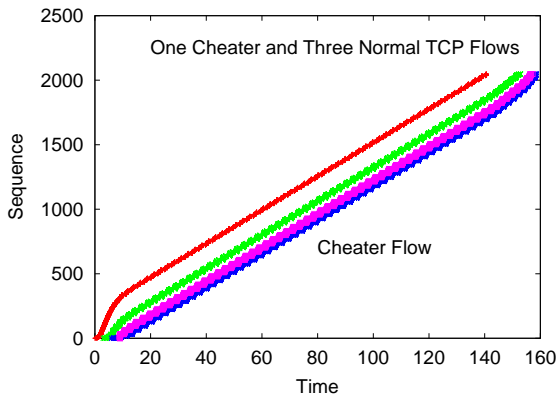Figure 5: Experiment 2, One Cheater

Figure 3: Experiment 1, One Cheater
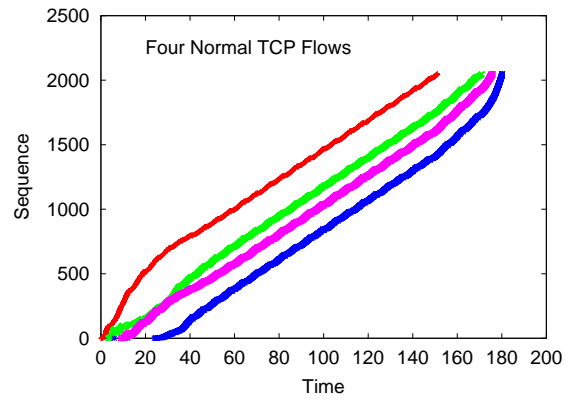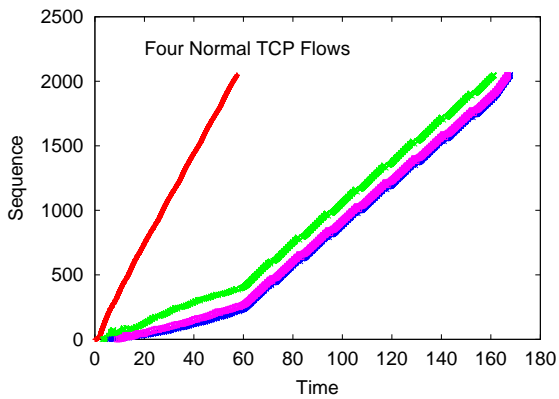

Figure 6: Experiment 3, Four Normal TCP

Figure 4: Experiment 2, Four Normal TCP
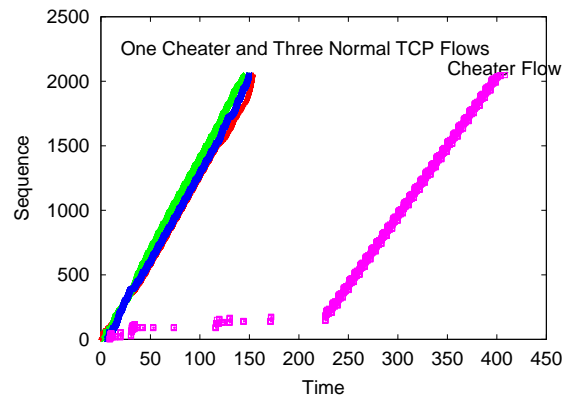
Figure 7: Experiment 3, One Cheater

Note: Link lengths are not to scale

100 Mbps

30 Mbps Link
(bottleneck)

Randomly Distributed
56 Kbps – 1.2 Mbps Link
(bottleneck)

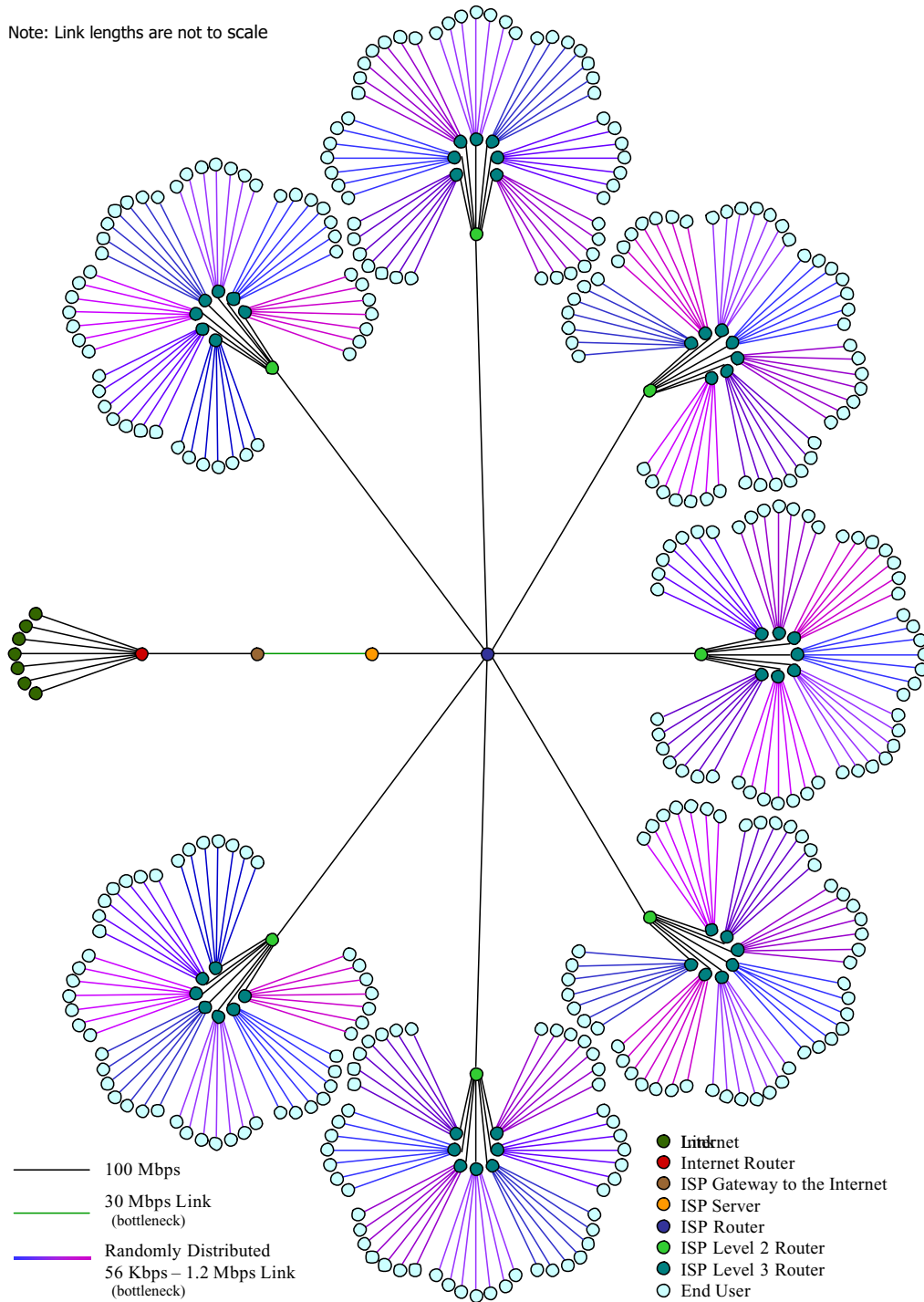Link Internet
Internet Router
ISP Gateway to the Internet
ISP Server
ISP Router
ISP Level 2 Router
ISP Level 3 Router
End User

Figure 8: Sample Study Topology

Table 1: Red vs. DropTail Results

	RED	DropTail	Tie
Best	4	4	3

Table 2: TCP Variations Results

	Tahoe	Reno	NewReno	SACK
Best	0	0	0	11

Although the conclusion of the groups that the Selective Acknowledgment feature of TCP produced the best results might have been influenced by the instructor's stated belief during class that TCP/*SACK* would likely exhibit the best performance. However, choosing which of the TCP variations is second best again gave no clear answer. As can be seen in Table 3, there is a near equal split between Tahoe, Reno, and NewReno.

Table 3: TCP Second Best

	Tahoe	Reno	NewReno	SACK
2nd Best	4	4	3	0

Looking at the results in more detail, Table 4 shows a representative set of comparative results for the four TCP variations. This particular set of results are for *DropTail* queues of varying lengths, and give the average response time (in seconds) for the simulated web browsing traffic loads (so smaller numbers represent better performance). These results represent the average of five different simulation runs, with different random number seeds for each. While this group did choose *SACK* as the "best" TCP variation, this is by no means a clear indication. The NewReno variation in fact performs better in this experiment for smaller queue sizes, up to about 200 packets. For larger queue sizes, *SACK* does appear to have a small advantage. We did not ask students to calculate standard deviations of results nor to produce confidence intervals, but we suspect that these results do not demonstrate a statistically significant performance advantage for any one TCP variation.

Table 4: TCP Variations, Response Time

QSize	Tahoe	Reno	NewReno	SACK
10	1.209	1.260	1.201	1.505
25	1.080	1.107	1.081	1.178
50	1.063	1.076	1.064	1.102
100	1.052	1.044	1.045	1.055
150	1.040	1.040	1.033	1.034
200	1.026	1.040	1.037	1.031
225	1.039	1.038	1.039	1.031
240	1.034	1.030	1.036	1.030
300	1.039	1.039	1.038	1.027
400	1.028	1.034	1.037	1.036

3 TOPOLOGY TAXONOMY

We have shown in the previous two sections the difficulty in determining clear winners in simulation based studies of network protocol performance. The variations in initial conditions, and the interaction between protocol performance and network configuration parameters lead to variations in measured results between different simulation scenarios. This leads us to the need for a large database of simulation scenarios, designed to represent a broad spectrum of network conditions. These scenarios should be created using a systematic approach to insure that they give complete coverage of likely network conditions encountered in large scale networks, such as the Internet. In this section, we outline a taxonomy of the variables and conditions that would be needed to create such a set of simulation scenarios, and give suggested range of values for each.

1. Source to Destination Hop Count. A number of previous studies have measured empirically the distribution of hop counts from a source to a destination in the Internet. One study (Riley, Ammar, and Zegura 2001) produced the hop–count histogram shown in Figure 9. Clearly, from this study, the set of topologies should include hop counts ranging from about 5 hops up to about 30 hops to give a representative sampling of hop counts found in the Internet.
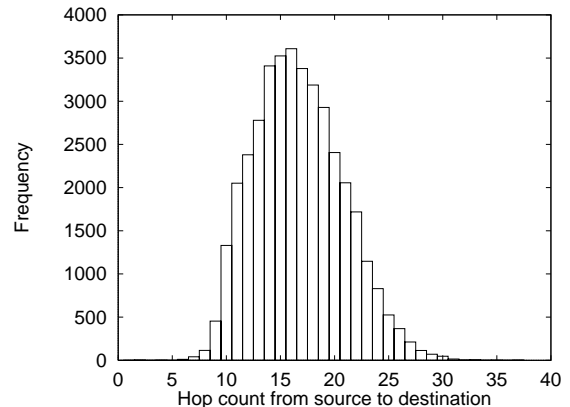

Figure 9: Hop Count Histogram

2. Round Trip Times. A distribution of empirically measured Internet round–trip–times (RTT's) is given in (Huffaker et al. 2002). This study indicated a tri-modal distribution for Internet RTT's, with modes at 30ms, 50ms, and 150ms; plus a large number greater than 300ms. Our set of standardized topologies must include similar distributions for RTT's, with a number of flows using experiencing RTT values in the four ranges mentioned above.

3. Link Bandwidth. Flows on the Internet encounter a wide variation in link bandwidth. Realistic values for first–hop end–user bandwidth values include 56Kb dial–up modems, 1.5Mb Cable and DSL modems, 10Mb Ethernet LANs, and 100Mb Ethernet LANs. Border gateway routers and network backbone routers have link bandwidths ranging from T1 (1.5Mb) to OC192 (about 10Gb). Our standardized topologies will have flows traversing a number of links with these range of bandwidth values.

4. Number of Congested Links. Typical simulation studies use the well–known *dumbbell* topology, with a number of flows sharing a single bottleneck link. Comparative studies of protocol performance must use a variable number of bottleneck links. Our topologies will include flows with as few as zero bottlenecks, and as many as three bottlenecks per flow.

5. Congested Link Cross–Traffic. Another failing of the *dumbbell* topology model is the lack of *Cross–Traffic* at congested links. In this context, cross–traffic is defined as two or more flows that share a congested link, but take divergent paths elsewhere in the network, potentially encountering more congestion. Our topology models will insure that for flows encountering multiple congestion points, a fraction of them take different paths with different congestion points.

6. Buffering Capacity on Congested Links. The maximum queue length at congested links has major impact on the overall performance of TCP. As was demonstrated earlier in section 2.1, queue capacities either too small or too large can have dramatic impact on relative performance. A commonly used rule of thumb is to set buffer limits to be some multiple of the delay–bandwidth product for the associated output link (Villamizar and Song 1994; Morris 1997), but there is no clear agreement on the multiplicative factor to be used. Our set of topologies will include a wide range of buffer limits, ranging from extremely small (10 packets) to extremely large (2000 packets).

7. Queuing Discipline on Congested Links. There have been a large number of proposed methods for queuing packets on congested links, but presently only two (*DropTail* and *RED*) have sufficiently widespread deployment to be included in comparative studies. Our topologies will include a mix of *DropTail* and *RED*, with some flows having one or the other exclusively and some flows encountering a mix.

8. Congested Link Offered Load. Congested links have degrees of congestion, ranging from buffer full conditions very occasionally to continual full conditions. Our topologies will generate congested links with a variety of congestion levels, from rarely congested to always congested.

9. Competing Traffic Characteristics. Flows encounter competing traffic at congested links that exhibit a wide variety of behavior. There are long–lived TCP flows (such as bulk data transfer applications) that have reached an equilibrium in the congestion–avoidance mechanisms; short lived *web–mice* that never get out of slow–start mode; and non–conforming UDP traffic with little or no congestion avoidance or control mechanisms, just to name a few. Our simulation topologies will include these and other competing traffic characteristics.

The scenarios given in this section are not intended to be an exhaustive list of the standardized topologies, but rather to outline a systematic approach to defining a realistic set of simulation scenarios that can lead to conclusive and meaningful results for simulation–based networking research.

4 CONCLUSIONS AND FUTURE WORK

Making comparative claims about network performance based on simulation studies is indeed a risky endeavor. Small and seemingly insignificant changes in network parameters can have substantive affects on measured performance, leading to incorrect or inconclusive results. By taking a systematic approach, defining a large set of simulation scenarios that cover a number variations in network conditions, one can study network behavior with more confidence.

As future work, we will create a set of simulation scenarios, based on the popular *ns2* (McCanne and Floyd 1997) network simulator, that gives good coverage for the various network parameters discussed previously. This set of topologies will be made freely available to network researchers, allowing more thorough and detailed studies of network performance.

However, even when the exhaustive set of simulation scenarios has been created, there is further research required to determine how to combine and interpret results from a large set of experimental simulations. We expect that no protocol will be the clear "winner" for every single scenario, but rather would exhibit better performance in a plurality of the cases. We intend to investigate ways to realistically assign weights to the sets of results such that scenarios that are *likely* are weighted more heavily than results from less likely scenarios.

REFERENCES

Bennett, J. C. R., C. Partridge, and N. Schectman. 1999, Packet reordering is not pathological network behavior. *IEEE Transactions on Networking* 7 (6): 789–798.

Bruyeron, R., B. Hemon, and L. Zhang. 1998. Experimentations with tcp selective acknowledgment. *Computer Communication Review* 28 (2): 54–77.

Christiansen, M., K. et al. 2000, Tuning RED for web traffic. In *Proceedings of ACM SIGCOMM 2000*, 139–150.

Fall, K., and S. Floyd. 1996, Simulation-based comparisons of Tahoe, Reno, and SACK TCP. In *Computer Communications Review*.

Floyd, S., and T. Henderson. 1999, Internet RFC2582: The newreno modification to tcp's fast recovery algorithm. Network Working Group.

Floyd, S., and V. Jacobson. 1993, Random early detection gateways for congestion avoidance. *IEEE Transactions on Networking* 1 (4): 397–413.

Huffaker, B., D. Plummer, D. Moore, and K. Claffy. 2002. Topology discovery by active probing. In *Proceedings of the 2002 Symposium on Applications and the Internet (SAINT'02)*, 90–96.

Jacobson, V. 1988, August. Congestion avoidance and control. In *Computer Communications Review, Proceedings of ACM SIGCOMM 88*.

Mathis, M. et al. 1996, Internet RFC2018: Tcp selective acknowledgement options. Network Working Group.

McCanne, S., and S. Floyd. 1997. The LBNL network simulator. Software on-line: [http://www.isi.edu/nsnam]. Lawrence Berkeley Laboratory.

Morris, R. 1997, TCP behavior with many flows. In *Proceedings of the 1997 Conference on Network Protocols (ICNP'97)*.

Riley, G. F., M. H. Ammar, and E. W. Zegura. 2001, Efficient routing using nix-vectors. In *2001 IEEE Workshop on High Performance Switching and Routing*.

Villamizar, C., and C. Song. 1994, High performance tcp in ansnet. In *Computer Communications Review*, 45–60.

ACKNOWLEDGMENTS

AUTHOR BIOGRAPHY

GEORGE F. RILEY is an Assistant Professor of Electrical and Computer Engineering at the Georgia Institute of Technology. He received his Ph.D. in computer science from the Georgia Institute of Technology, College of Computing, in August 2001. His research interests are large–scale simulation of computer networks, and networking research using these simulation techniques. He can be reached via email at <riley@ece.gatech.edu>.