Empirical Models of TCP and UDP End–User Network Traffic from NETI@home Data Analysis

Charles R. Simpson, Jr., Dheeraj Reddy, George F. Riley
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
{rsimpson,dheeraj,riley}@ece.gatech.edu

Abstract

The simulation of computer networks requires accurate models of user behavior. To this end, we present empirical models of end–user network traffic derived from the analysis of NETI@home data. There are two forms of models presented. The first models traffic for a specific TCP or UDP port. The second models all TCP or UDP traffic for an end–user. These models are meant to be network–independent and contain aspects such as bytes sent, bytes received, and user think time. The empirical models derived in this study can then be used to enable more realistic simulations of computer networks.

1. Introduction

The simulation of computer networks has become a popular method to evaluate characteristics of these networks across a wide range of topics, including protocol analysis, routing stability, and topological dependencies, to name a few. However, for these simulations to yield meaningful results, they must incorporate accurate models of their simulated components.

One such component is end–user traffic generation. This component should be network–independent so that it can be used in a wide variety of simulation configurations without dependency on the simulated environment. These traffic models should be updated frequently, using recent measurements, to accurately reflect the changing nature and uses of the Internet. Further, such measurements should represent the heterogeneous connection methods and diverse locations of Internet users. To this aim, we have developed network–independent traffic models for network users based on data gathered by the NETI@home infrastructure.

The remainder of this paper is organized as follows. Section 2 presents work related to this study. Next, Section 3 describes the dataset used for this study and the methodology used to create our models. Section 4 discusses the experimental results of our study and Section 5 describes the simulation used to demonstrate and validate our models. Finally, Section 6 discusses several areas of future work and we conclude in Section 7.

2. Background and related work

Portions of this work are based on work presented in [13] and [17] and we have chosen to adopt much of their nomenclature. However, we have attempted to expand upon their work in several ways. First, the work in [13] is based on packet traces collected from a campus network. In an attempt to represent more typical end–users, we use data collected by the NETI@home project. Also, the studies conducted in [13, 17] were specific to TCP connections on port 80. In this study, we model any given TCP or UDP port, as well as all TCP or UDP traffic aggregated.

NETI@home[16] (Network Intelligence at home) is an open–source software package named after the popular SETI@home[1] software. The NETI@home client is available on the NETI@home website[15] and is designed to be run by any client machine connected to the Internet. When run on a client machine, the NETI@home software reports end–to–end flow summary statistics to a server at the Georgia Institute of Technology. The statistics collected and the functionality of the software are discussed in [16]. Since NETI@home is designed to run on end–user systems, it provides a unique perspective into the behavior of both end–users and their systems.

Previously, NETI@home data analysis has focused on aspects relating to security[9]. In this paper, we utilize the measurements made by NETI@home to generate traffic models based on end–user behavior. NETI@home users represent a heterogeneous mixture of network users from various networks and geographical locations.

Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation (PADS'06)
0-7695-2587-3/06 $20.00 © 2006 IEEE

The need for accurate simulation models was discussed in [8]. Several other studies have discussed modeling of either application–specific [3, 4, 5, 6, 17] or general [2, 10, 11, 18] end–user network traffic. Also, several studies have used network traffic models in simulation environments including [7, 12, 19, 20].

3. Methodology

The models developed for this work are intended to be network–independent. To this aim, we define several characteristics of TCP and UDP flows that reflect this design choice and attempt to wholly represent network client behavior.

There are two categories of models created in this study. The first is specific to a TCP or UDP port, that is we create a model of client behavior for a given TCP or UDP port. Throughout most of this paper, we use the model created for TCP port 80, the most common port used by World Wide Web servers, as an example. The second category of model created is an aggregate of all port–specific models. This model can be likened to a TCP or UDP client model. Such a model may prove useful for studies that are more generic and are not attempting to study a particular type of network traffic. All of these models incorporate empirical distributions directly interpreted from the NETI@home dataset.

The dataset used in this study consists of NETI@home data collected over a one year period from October 1, 2004 to September 30, 2005. This dataset includes over 36 million TCP flows and 93 million UDP flows, which form the basis of this work, as well as various other flow types and information about their corresponding hosts. Although an exact calculation is not possible due to privacy settings and dynamically assigned IP addresses, we estimate that this data was collected by approximately 1700 users. These users represent a heterogeneous sampling of Internet users running some 8 different operating systems and reporting from approximately 28 nations and 43 US ZIP Codes.

The first two aspects we model are empirical distributions of *bytes sent* and *bytes received*. These values are based only on the payload of the packets and thus do not represent the sizes of the TCP or UDP headers and their underlying headers or TCP's flow control and congestion control algorithms, merely transferred application information. This allows our models to be used in simulations where variations of TCP or UDP are employed.

The next aspect modeled is *user think time*. User think time is the term we use for the amount of time a client waits before initiating another flow. For this aspect, we developed two empirical distributions. One distribution describes the user think time when consecutively accessing a specific destination and the other describes the user think time when contacting a new destination.

Another aspect modeled is *consecutive contacts*. Consecutive contacts is the term we use for the probability that a client will choose to initiate another flow with the last destination contacted, or the client will choose to initiate a flow with a new destination. For this aspect, we developed a single empirical distribution.

Finally, the last aspect modeled is *contact selection*. Contact selection is the term we use for the frequency distribution of contacting specific destinations. This distribution can be thought of as modeling the popularity of a destination. For this aspect, we developed a single empirical distribution.

One other aspect that we believe to be worth modeling is related to *idle time*. For applications such as World Wide Web transfers, this aspect has little meaning, as web pages are simply requested and served. However, for interactive applications such as SSH or telnet, there are periods of time, *during* the flow, when there is no data transferred. However, using the NETI@home data, it is difficult to differentiate between network–dependent flow time and network–independent flow time. We are aware of work [10, 11] that attempts to capture this behavior and are considering implementing a similar technique into the NETI@home client software so that future models can incorporate this aspect of user behavior.

4. Experimental results

From the analysis of the NETI@home dataset described previously, we were able to generate a set of empirical distributions for each component of our models. To download the complete set of distributions and for any updates to these distributions please visit `http://neti.gatech.edu/research/user.html`.

4.1. Bytes sent

The amount of bytes sent varies dependent on the port modeled. However, upon investigation of each modeled port, our findings seem intuitive.

Figure 1 depicts the cumulative distribution function of bytes sent for TCP port 80. Compared with previous studies [13], these results contain many more flows with zero bytes sent. However, upon investigation it does not appear that these results are due to a single NETI@home user or are anomalous. This difference in results is most likely due to the fact that [13] was based on data collected from a campus network, whereas NETI@home data contains users with less reliable network connections. The zero bytes sent flows typically represent flows in which the connection failed during the TCP three–way handshake. Although these flows do not generate much network traffic (usually no more than

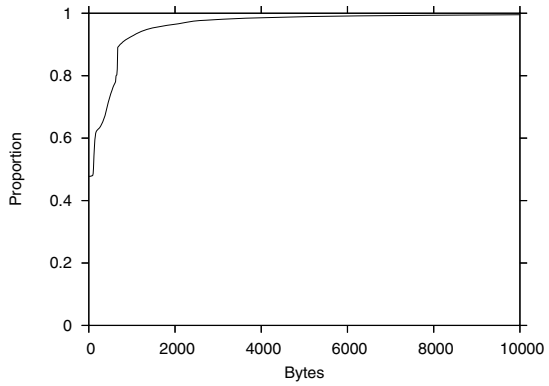


Figure 1. CDF of bytes sent for TCP port 80

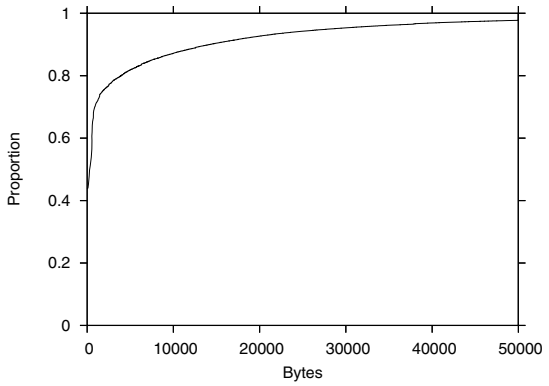


Figure 2. CDF of bytes received for TCP port 80

three packets), they are significant in terms of numbers of flows and most likely influence a user's behavior.

As can be seen in the figure, approximately 40 percent of flows to TCP port 80 send little or no data. There are several possible causes for the large number of flows sending little or no data. First, many of these flows are failed connection attempts. Many NETI@home users are utilizing less reliable network connections such as dial–up or wireless. Also, some of these flows may be to blocked sites. Many browsers and third–party software block advertisements and some organizations restrict the viewing of certain websites. Finally, a handful of NETI@home users periodically scan hosts on the Internet[9]. Considering that these users know that their network connections are monitored, it is unlikely that this scanning is intentional and may be the result of a virus or worm. While these results could be considered anomalous, we believe that this does indeed represent typical end–user behavior as seen on the Internet. Almost all remaining flows send no more than 10 KB of data to the server.

4.2. Bytes received

The amount of bytes received by the client is also dependent on the port modeled. Figure 2 depicts the cumulative distribution function of bytes received for TCP port 80. Compared with [13], we also find that there are many more flows with zero bytes received. As with our findings for bytes sent, this is most likely due to failed connection attempts.

The distribution for bytes received has a much longer tail than that for the bytes sent. Approximately 40 percent of flows with a remote TCP port of 80 receive little or no data. However, more than 10 percent of these flows receive greater than 10KB of data.

4.3. User think time

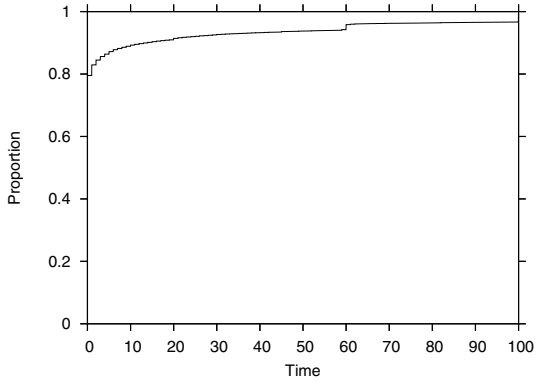
The cumulative distribution function for user think time to the same destination is given in Figure 3 and to differing destinations is given in Figure 4 for TCP ports 23 and 80. These findings show a tendency towards shorter user think times than was found in [13] for TCP Port 80. We can think of several reasons for this shortened user think time. First, the World Wide Web has become much more popular since the time of [13]'s publication. Also, it is likely that NETI@home captures data from users who are active more often than it does for inactive users as many users would simply turn off their machines while not using them, thus disabling NETI@home's monitoring. This would artificially inflate our numbers to show users that appear to be more active and is a source of bias.

We chose to model the user think time to the same destination separately from the user think time to a different destination. Figures 3(a) and 4(a) appear to be similar however. We believe that it is still appropriate to model these think times separately as these distributions can differ greatly for other TCP or UDP ports as is shown in Figures 3(b) and 4(b). These figures show the distributions for think times for TCP Port 23, the port commonly used for telnet.

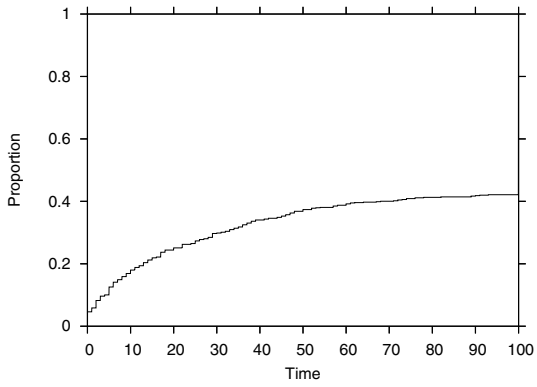
For connections to TCP port 80, the majority of user think times tends to be less than 1 second. However, for connections to TCP port 23 (telnet), the user think times have a much heavier tail, with only approximately 40 percent of flows having think times less than 100 seconds.

4.4. Consecutive contacts

In Figure 5, we present the cumulative distribution function for consecutive contacts for TCP port 80. These results also show a tendency towards a lower number of consecutive contacts than was found in [13]. However, this is

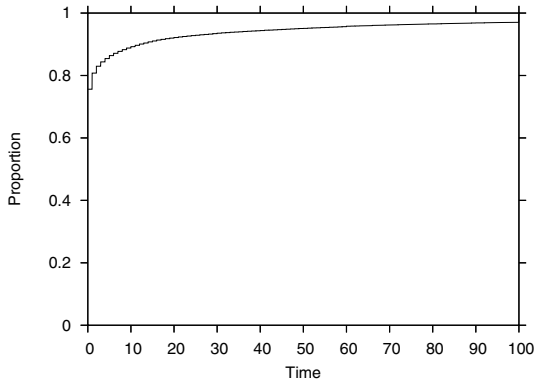


(a) TCP port 80

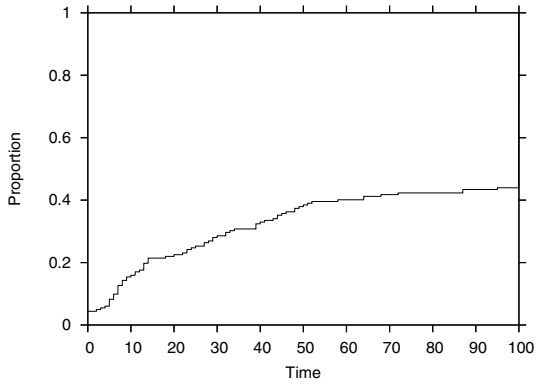


(b) TCP port 23

Figure 3. CDF of user think time to same IPs



(a) TCP port 80



(b) TCP port 23

Figure 4. CDF of user think time to differing IPs

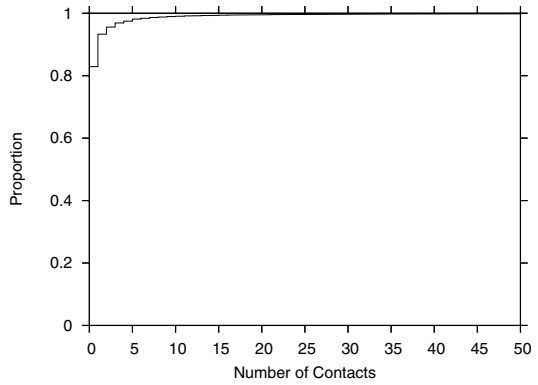


Figure 5. CDF of number of times an IP is contacted consecutively for TCP port 80

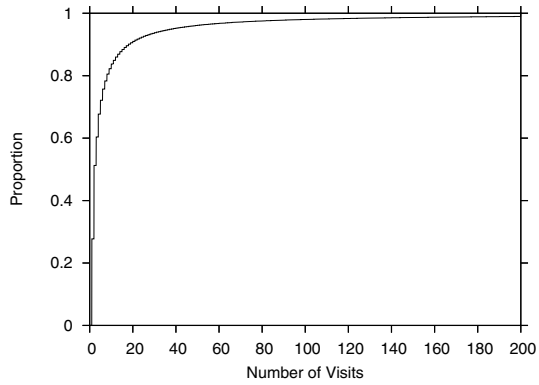


Figure 6. CDF of relative frequency of server visits for TCP port 80 over a one year period

intuitive considering the number of "failed" connection attempts observed previously.

Approximately 80 percent of the flows to TCP port 80 are not consecutive, that is the destination is contacted only once in a row. Further, over 99 percent of visits to a specific destination on TCP port 80 lasted for 10 or less flows in a row. Therefore, it appears that users tend to switch web destinations fairly often as was noted in [13].

4.5. Contact selection

Unlike [13], which used a Zipf distribution, we were able to construct a cumulative distribution function for contact selection due to the wide sampling offered by the NETI@home dataset. Figure 6 presents this CDF for TCP port 80. One possible source of inaccuracy for this aspect is the fact that we are unable to determine if a specific destination uses multiple IP addresses, thus reducing the frequency of selection a given contact may appear to have.

As can be seen in the figure, for TCP port 80 servers the distribution of the overall number of visits by NETI@home users is quite varied and has a heavy tail. Many servers are only visited a handful of times, however many other servers tend to be contacted quite often, with some servers receiving millions of visits over the year studied.

5. Simulation results

To judge the usefulness of our models, we have incorporated the above derived TCP traffic models into the GTNetS environment[14]. The GTNetS environment already has some HTTP traffic models as described in [13]. We incorporated the models derived from the analysis of the NETI@home datasets into GTNetS. We consider this approach to be a better one for traffic generation in network simulations, because NETI@home datasets are more current and continue to be so [16]. An analysis program generates these models automatically from the NETI@home datasets. The traffic distribution models can then be easily used by the application layer models which drive a network simulation. In our simulation experiments, we have concentrated on the World Wide Web traffic and the HTTP models. Our implementation samples the empirical distributions to determine the particular values used at a given time. This seems a logical choice since any single distribution doesn't seem to fit the complete dataset verifiably. We model the behavior of a web browser in GTNetS which sends a HTTP request to a designated webserver asking it to send a certain length of data that constitutes the response. When the simulation starts, the browser application chooses a server randomly from a list of target servers. It then chooses a *response size* that it wants to obtain from the webserver from the CDF that describes the *received bytes*. The size of the HTTP request packet is chosen from the *sent bytes* CDF plot. It may request one or more objects within the same TCP connection. Once the web browser application has received the appropriate response, it proceeds to select a different server or the same server for its next request and waits for an amount of time. This amount of time, which is obtained from the CDF that describes the *user think time*, depends on whether the same server is chosen or a different server is chosen.

The network topology for simulations is obtained from [7]. It consists of a large set of web browsers connected via a series of three routers to a webserver as shown in Figure 7. We have chosen this to be our baseline topology because we have earlier simulation experiments conducted using the models and datasets proposed in [7].

The simulation experiment is run using two HTTP traffic models. One of the traffic models is obtained from the datasets suggested in [13] and [7]. The other traffic model

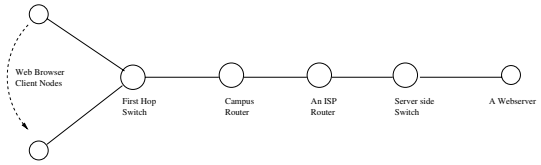


Figure 7. Network topology used for testing traffic models in simulation

Table 1. Variation in average and maximum response times when using HTTP traffic model presented in this paper

Number of browsers	Average response time	Maximum response time
10	0.316738	0.738639
25	0.301151	0.740423
50	0.318433	0.738642
75	0.321075	0.743916
100	0.304644	0.745433
125	0.305372	0.751632
150	0.312204	0.839426

is one that is obtained from the NETI@home datasets. Intuitively, empirical traffic models should be more representative of a realistic dataset than statistical traffic generators, although the former cannot be subjected to extrapolations. All the measurements are the averages of three runs of a simulation at a given data point.

Table 1 shows the average and maximum response times for a given number of web clients when they request data from a webserver using the traffic models presented in this paper. Table 2 shows the average and maximum response times for the same number of web browsers when they request data from a webserver using the traffic models presented in [13].

It can be seen from the results in Table 1 and Table 2 that the maximum response time for the HTTP traffic model presented in [13] is substantially larger than the model that is derived from NETI@home dataset. A careful observation of the cumulative distribution functions of the two datasets shows that the NETI@home data has a larger proportion of flow sizes that are very small, most likely due to the inclusion of a large number of failed connections. This results in lower load on the webserver and consequently lower latencies. This is evident in the lower average and maximum response times as the traffic increases. On the other hand, the traffic model presented in [13] has a lesser number of flow sizes that are very small. This results in a larger load

Table 2. Variation in average and maximum response times when using HTTP traffic model presented in [13]

Number of browsers	Average response time	Maximum response time
10	0.461172	0.716738
25	0.339998	1.01388
50	0.344094	1.20155
75	0.375188	3.98217
100	0.380281	3.80786
125	0.332889	4.16023
150	0.405156	6.6588

on the server and on the network as the number of web browsers increases. When the number of web browsers is fairly small, the difference is not appreciable because the flow size does not influence the network.

The code used for these simulations, as well as the empirical models, are available in the latest official distribution of the GTNetS environment.

6. Future work

Several enhancements to our modeling technique can be made and are areas of future work. First, it would be useful to model idle times within a flow. As previously mentioned, certain applications have periods of time where the connection is idle as in interactive applications. Another enhancement to our model would be to determine if there is any correlation between the different aspects of our model. For example, in certain applications the number of bytes sent and the number of bytes received may be highly correlated. If so, these aspects should most likely be treated as bivariate data. Several enhancements could also be made to our consecutive contacts and contact selection components. It is intuitive that once a destination is visited and then left, that the original destination has a higher likelihood of being visited again. Thus, a model with memory, such as a Markov model, would be useful. Such a model may also incorporate zero byte flows. That is, if a connection fails, the likelihood of that connection's destination of being visited again may change. Further, our model could be extended to other protocols beyond TCP and UDP. Currently, NETI@home collects flow summary statistics for TCP, UDP, ICMP, and IGMP, so ICMP and IGMP models could easily be derived.

The models presented in this paper solely focus on network–independent characteristics. It would be useful however to model network–dependent aspects of the global Internet. Such a model could focus on parameters such as

the proliferation of network address translation, the topology of the Internet, the number of servers visited overall, latency, loss, bandwidth, and the locality of network traffic.

The nature of the Internet and its usage is constantly changing. With an infrastructure such as NETI@home in place, changes to Internet usage, and thus updates to our models, should be studied. This will not only allow for studies comparing changing trends, but will ensure the availability of accurate and updated simulation models.

Finally, we have chosen to represent our models in empirical form. Such a form has its advantages, however analytical models could be developed from this data. These analytical models may have advantages for scaling, both temporally and spatially.

7. Conclusions

In conclusion, we have presented empirical models of end–user network traffic. There are two general forms of these models, one form is port–specific for a given TCP or UDP port. The second form is a generic model for TCP or UDP traffic. These models consist of network–independent distributions for the number of bytes sent, the number of bytes received, the user think time to the same destination, the user think time to a different destination, the number of times a destination will be contacted consecutively, and the popularity of specific destinations.

The distributions derived are based on the NETI@home dataset and are meant to represent a heterogeneous sampling of network users. Such a heterogeneous sampling of users from differing network and geographical locations provides more accurate models for simulations. As the NETI@home project is ongoing for the foreseeable future, we plan to continuously update the models. For these updates and to download the complete distributions please visit `http://neti.gatech.edu/research/user.html`.

Further, we have implemented these models in a simulation environment. In this simulation environment we tested the affect of network traffic on a webserver. These results were then compared to the results from previous models. The models and code used are available in the latest distribution of the GTNetS environment.

References

[1] D. P. Anderson and et al. SETI@home: Search for extraterrestrial intelligence at home. Software on-line: http://setiathome.ssl.berkeley.edu, 2003.

[2] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski. Modeling internet backbone traffic at the flow level. *IEEE Transactions on Signal Processing – Special Issue on Networking*, 51(8), August 2003.

[3] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *ACM SIGMETRICS*, 1998.

[4] J. Cao, W. S. Cleveland, Y. Gao, K. Jeffay, F. D. Smith, and M. C. Weigle. Stochastic models for generating synthetic HTTP source traffic. In *IEEE INFOCOMM*, March 2004.

[5] Y.-C. Cheng, U. Holzle, N. Cardwell, S. Savage, and G. M. Voelker. Monkey see, monkey do: A tool for TCP tracing and replaying. In *Proceedings of USENIX Technical Conference*, June 2004.

[6] H.-K. Choi and J. O. Limb. A behavioral model of web traffic. In *ICNP*, 1999.

[7] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith. Tuning RED for web traffic. *IEEE/ACM Transactions on Networking*, 9(3):249–264, June 2001.

[8] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, August 2001.

[9] J. B. Grizzard, C. R. Simpson, Jr., S. Krasser, H. L. Owen, and G. F. Riley. Flow based observations from NETI@home and honeynet data. In *Proceedings from the sixth IEEE Systems, Man and Cybernetics Information Assurance Workshop*, pages 244–251, June 2005.

[10] F. Hernandez-Campos, A. B. Nobel, F. D. Smith, and K. Jeffay. Understanding patterns of TCP connection usage with statistical clustering. In *IEEE MASCOTS*, 2005.

[11] F. Hernandez-Campos, F. D. Smith, and K. Jeffay. Generating realistic TCP workloads. In *Computer Measurement Group International Conference*, December 2004.

[12] L. Le, J. Aikat, K. Jeffay, and F. D. Smith. The effects of active queue management on web performance. In *ACM SIGCOMM*, pages 265–276, August 2003.

[13] B. A. Mah. An empirical model of HTTP network traffic. In *IEEE INFOCOMM*, April 1997.

[14] G. F. Riley. The Georgia Tech Network Simulator. In *Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, pages 5–12, 2003.

[15] C. R. Simpson, Jr. NETI@home. Software on-line: http://neti.gatech.edu, 2003. Georgia Institute of Technology.

[16] C. R. Simpson, Jr. and G. F. Riley. NETI@home: A distributed approach to collecting end-to-end network performance measurements. In *PAM2004 - A workshop on Passive and Active Measurements*, April 2004.

[17] F. D. Smith, F. Hernandez-Campos, K. Jeffay, and D. Ott. What TCP/IP protocol headers can tell us about the web. In *ACM SIGMETRICS*, pages 245–256, 2001.

[18] J. Sommers, H. Kim, and P. Barford. Harpoon: A flow–level traffic generator for router and network tests. In *ACM SIGMETRICS*, June 2004.

[19] M. Weigle, K. Jeffay, and F. D. Smith. Delay–based early congestion detection and adaptation in TCP: Impact on web performance. *ACM Computer Communications Review*, 28(8):837–850, May 2005.

[20] J. Xu and W. Lee. Sustaining availability of web services under distributed denial of service attacks. *IEEE Transactions on Computers*, 52(2):195–208, February 2003.