# Building a Network Simulation Model of the Teragrid Network

Thomas Hacker[†‡] Preston Smith[*†]

†Computer & Information Technology, ‡Discovery Park Cyber Center,
*Rosen Center for Advanced Computing

Purdue University, West Lafayette, IN

———————————— ◆ ————————————

## 1 Introduction

Global research teams are building large-scale cyberinfrastructure systems, which couple computing, storage, data publishing, and collaboration tools into a platform to support scientific collaboration. Modern scientific research is increasingly powered by simulation and analysis that combines high performance computers with terabytes of data. Poor performance and low reliability of these systems can seriously impact research projects that rely on cyberinfrastructure. Performance problems include unacceptably poor network performance for applications that transmit data over wide-area high speed networks, and poor reliability and usability of cyberinfrastructure systems caused by faults and performance problems induced by the poorly tested interaction of applications, middleware, and infrastructure. Thoroughly testing cyberinfrastructure components before deployment is essential to prevent many of these problems. Unfortunately, testing these components on a production infrastructure is difficult and potentially disruptive. In this paper, we describe our work in designing and using an accurate network simulation model of the Teragrid network, and the performance of a number of high speed TCP variants. We introduce Stochastic TCP, a new TCP congestion avoidance approach that improves performance over high speed wide area networks for applications that transfers large amounts of data, yet retains critical fairness characteristics that are necessary to prevent congestion collapse.

## 2 Need for a simulation testbed

Achieving acceptable levels of performance on high speed wide area networks is difficult, and cyberinfrastructure applications that rely on fast data transmission are seriously impacted by slow and unreliable network connections. New technologies are continuously developed to address performance and reliability problems, but thoroughly testing them under realistic conditions is challenging. Large shared infrastructure projects that support a large research community, such as the Teragrid, are vulnerable to problems arising from unforeseen interactions among infrastructure, middleware, and applications.

One example that illustrates these problems is efforts by the networking research community to increase performance through the development and use of aggressive network protocols. Although effective at increasing performance, many new approaches to improve the performance of TCP congestion avoidance have a high probability of unfairly stealing bandwidth from other users and affecting production systems, or in the worst case inducing congestion collapse for all of the network traffic sharing the network.

Thoroughly testing these components on a production infrastructure is not practical, and is problematic for several reasons: i) testing may disrupt critical production work; ii) it is difficult to partition and reserve large parts of the infrastructure for dedicated performance experiments and testing; iii) the irreproducible nature of background load on networks and infrastructure complicates experimental design; and iv) it is not possible to substantially reconfigure the systems and networks for testing.

Experimental testbeds that can accurately simulate and emulate deployed infrastructure are critical to support the development of new cyberinfrastructure technologies. Accurately modeling wide-area high-speed networks is difficult – many of the parameters that affect network performance, such as latent packet loss, are poorly understood. In this paper, we describe a network simulation framework we developed that accurately characterizes the Teragrid network. We describe our experiences using this testbed with the a large institutional grid computing infrastructure at Purdue University to develop a new high speed TCP protocol that improves end-to-end network performance for data intensive applications.

## 3 Developing a network simulation model

Many network simulation and emulation tools are available that can be used to create detailed and realistic testing and assessment conditions for network applications. These tools include ns-2 [1], NISTnet [2], dummynet [3], and OPNET [4]. Although these tools can be used for assessing new protocols and applications, it is not clear how to precisely configure and instrument these tools to accurately replicate the most critical aspects of the infrastructure.

Aware of these problems, we decided to develop a testbed to emulate the Teragrid network for the purpose of testing new networking protocols. Our aim was to create a realistic network simulation and emulation that would allow us to thoroughly test and assess a new version of high speed TCP named Stochastic TCP [5] under realistic network conditions without affecting the production Teragrid infrastructure. The goals of this effort were several fold. First, we wanted to create a simulation and emulation testbed featuring a network topology that is similar to the Teragrid network. Second, since TCP is very sensitive to packet loss, which arises from both congestion and non-congestion sources [6], we decided to develop and use a detailed and accurate model for packet losses based on measured network characteristics. Third, our network topology used link speeds between routers and systems based on actual Teragrid network link speeds. Using this approach, we could vary link speeds and compare results to the existing network configuration to assess the effects of changes on performance. Fourth, we wanted the ability to inject network traffic into the topology at various points to assess the effects of applications and cross-traffic on performance and reliability. Finally, we wanted to be able to thoroughly test Stochastic TCP under a wide variety of network conditions and background traffic loads.

Using this testbed, we tested a large variety of TCP variants under a wide range of conditions, and assessed their effectiveness and fairness when competing with standard TCP. The next section of this paper describes the process and the tools we used to develop this testbed.

## 4 Methods

To develop an accurate testbed model of the Teragrid network, we first had to understand the salient features of the network that would have the greatest effect on the testbed. Some of the most obvious characteristics, such as the network topology and link speeds, could be discovered using tools such as *traceroute* and *ping.* Other characteristics, such as background packet loss and host connections, took more work to define. In this section, we describe the process followed in discovering, developing, and validating these critical characteristics of the Teragrid network.

### 4.1 Network Topology

The Teragrid network is a dedicated high performance wide area network connecting Teragrid resource providers across the United States. The list of sites involved in the Teragrid includes SDSC, LONI, NICS, NCAR, Argonne, NCSA, IU, Purdue, PSC, TACC, and ORNL. Each resource provider is connected to the Teragrid network via a dedicated 10 Gb/sec network link.

The most critical aspects of the Teragrid network that affect performance in our network simulation model are: network topology, host connections, link speeds and latencies, loss models, queue length, and the simulated applications that drive network simulations. The primary features of the network topology are the speeds of the links between end hosts and routers on the network path between a sender and receiver, and transmission latencies between individual nodes on the network. To determine the values for these parameters for the Teragrid network, we used *traceroute* along with information provided by the Teragrid, Purdue, and CALREN Network Operations Centers to develop a simulation network topology. Figure 1 shows the network topology we developed for ns-2 simulations. This topology uses link speeds and latencies based on the salient characteristics of the Teragrid network. Several routers on the Teragrid backbone network are connected by four 10 Gb/sec links, which are individual 10 Gb/sec lambdas on the optical path between routers. For our study, we determined that simulating only one 10 Gb/sec lambda between backbone routers was sufficient for characterizing TCP performance and behavior on the network without complicating the simulation framework, especially since the aggregate throughput from sources would not exceed 10 Gb/sec at any bisection of the network topology. We derived latencies between nodes using the latencies reported by *traceroute.* We simulated several hosts at SDSC and Purdue connected to the edge routers with dedicated 1 Gb/sec links, and added intermediate data sources and sinks within the network topology as hosts connected by 1 Gb/sec links. The addition of these crosstraffic nodes allows us to study the effects of network data transmission between SDSC and Purdue on cross-traffic performance and latency.

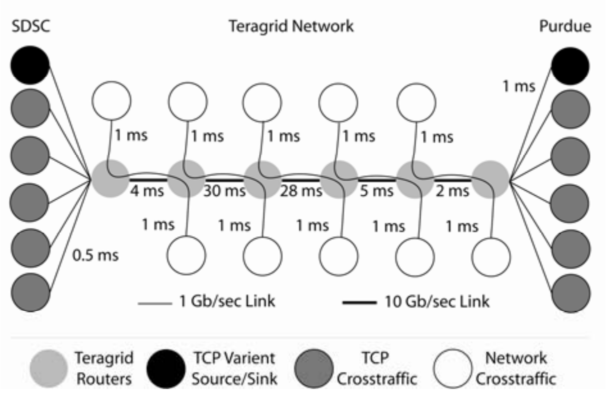

**Figure 1. Network Simulation Topology**

### 4.2 Network Simulation Loss Model

In addition to link speeds and latencies, a critical element of an accurate network simulation is a realistic statistical model of packet losses. In practice, the core network experiences very few packet loss events that arise from overloading or congestion of the network. Most problems seem to occur on the end hosts, or the local and campus area networks connecting these hosts to the core networks.

TCP congestion avoidance is very sensitive to packet loss, and the performance characteristics of TCP are strongly influenced by the statistical behavior of packet loss. An accurate simulation of the network must include a realistic model of packet losses. To instrument the simulation with an accurate statistical loss model, we performed a series of network transfer experiments from San Diego Supercomputer Center to Purdue University over the Teragrid network. We transferred approximately 1 TB of data (114 million packets) between 8pm on June 19, 2007 to 2am on June 20, 2007. During this test period, the network operations centers at SDSC, Teragrid, and Purdue confirmed there were no packets losses due to network congestion logged for routers on the network path during the time of the transmission experiment. We collected a packet trace of the transfer using *tcpdump*, modified *tcptrace* [7] to identify packet losses from the packet trace, and calculated the elapsed time between loss events. To determine the statistical distribution of the elapsed time between losses we used the EasyFit package from MathWave [8], and found that a unimodal Weibull distribution was the closest fit based on the Anderson-Darling test. A typical statistical model used to generate time between packet losses for the ns-2 simulator is based on an exponential distribution of the elapsed time between packet loss events. We found that the Weibull distribution was a closer fit than to the observed time between losses then the exponential distribution, and that the use of a Weibull distribution in place of the exponential distribution as the underlying statistical driver of loss events resulted in a more accurate loss model. The Weibull parameters for the closest statistical distribution fit to actual packet losses observed during our tests were shape parameter 2.9 and scale parameter 0.3. Ns-2 does not support a Weibull error model, so we extended the ns-2 exponential loss model to support a configurable Weibull based statistical packet loss model. The number of packets transmitted or time elapsed between loss events in the loss model was selected from the Weibull statistical distribution described by the equation

$$G(t) = 1 - e^{(-\lambda t)^{\alpha}}$$

where $\beta = \frac{1}{\lambda}$ is the scale parameter, and $\alpha$ is the shape parameter [9]. Note that this distribution is closely related to an exponential distribution: when $\alpha = 1$, G(t) is an exponential distribution.

To confirm the validity of our loss model and network simulation topology, we instrumented an ns-2 simulation using the Weibull loss model and the topology shown in Figure 1, and compared simulation results with real data transfers. Data transfer experiments over the real network path resulted in a throughput of 375.6 Mb/sec. To assess the accuracy of the simulation, 83 single TCP SACK stream transfers were simulated with a transfer time of 1000 seconds. The throughput of the simulated transfers was 451.45 Mb/sec with a standard deviation of 103 Mb/sec. The closest fitting statistical distribution to the simulation data was a Wakeby distribution, with a median value of 375 Mb/sec. The simulation results were similar enough to the transfer experiments to convince us that the instrumentation of our simulation would be an adequate testbed for our experiments.

Packet losses arise from two sources: network congestion, in which overloaded routers and switches are forced to drop packets when the rate of incoming data exceeds the maximum supported output rate; and non-congestion sources of loss that cause drops, which include hardware and software defects, overloaded end hosts, bad cables, the unexpected interaction of connected devices and software, and many other sources. To simulate the effects of non-congestive sources of packet loss and the combination of loss due to congestion and non-congestive sources, we developed two simulation scenarios. The first scenario used long interface queues of 1000 packets in the simulated router nodes, which were combined with an ns-2 statistical loss model that simulated non-congestion packet loss. The router queues were large enough to absorb any network transmission bursts and to prevent any packet loss from network congestion. The second scenario used short interface queues of 50 packets along with the ns-2 statistical loss model. This scenario generated losses from router packet drops when the queues overflowed, as well as packet losses from non-congestion sources using the ns-2 statistical packet loss model.

### 4.3 Simulated Application

The final critical element in creating a realistic network simulation is the application. The TCP congestion avoidance algorithm is a control system that is driven by data transmitted from the TCP sender to receiver. If an application creates short-lived TCP connections, or sporadically transfer small amounts of data, congestion avoidance will not be able to fully probe and utilize the available bandwidth. Thus, a simulated application that creates long-lived connections with a continuous stream of data to be transmitted is needed to fully load and test a TCP congestion avoidance variant. Ns-2 provides a simulated FTP application that meets these constraints, and we used this application in our ns-2 simulations. Each transfer experiment ran for 1,000 seconds of simulation time using a 9000 byte maximum transmission unit (MTU) packet size.

### 4.4 Running the Simulations

Running one simulated transfer experiment can take over 20 minutes to run to completion. For our simulations, we needed to vary several variables to fully explore the range of operational values and conditions, and to run enough trials for each set of parameters to produce statistically meaningful results. We estimated that we would need over

4,300 simulation trials, which would take over 3 months if we ran the simulations sequentially on a single processor.

To speed the time to completion, we used the Purdue BoilerGrid [10], which is one of the largest Condor installations deployed today with over 7,000 processors. Using the BoilerGrid, we were able to submit and complete several thousand simulations every day, which allowed us to quickly explore a broad simulation space and test the TCP variant as well as the simulation framework. Over a period of two months, we were able to perform over 30,000 detailed simulation trials using BoilerGrid, which provided very detailed and statistically significant and useful experimental results. Essentially, we were able to cast the simulation problem as a large embarrassingly parallel parameter sweep problem, using Condor to fully explore the simulation space.

We wanted to assess the performance of Stochastic TCP in competition with TCP streams from standard TCP Reno and SACK, as well as in competition with a wide variety of high speed TCP variants such as Scalable, CUBIC, and Hamilton. We used a set of ns-2 extensions developed by David Wei [11], and ported Stochastic TCP variant to ns-2 as a pluggable congestion avoidance model. We used a set of high speed TCP pluggable modules that were developed for David Wei's ns-2 extensions, specifically BIC, CUBIC, Fast, Compound, Reno, SACK, Hamilton, and Highspeed TCP.

## 6 Simulation Results

We assessed the performance of a number of TCP variants on the simulation Teragrid network. The variants we investigated were Stochastic TCP, BIC, Cubic, Fast, Compound, Reno, Sack, Hamilton TCP, Highspeed TCP, Scalable TCP, and Parallel TCP SACK streams.

We explored two simulation scenarios. The first simulated network transfers from a single node at SDSC to a single node at Purdue using FTP for a period of 1,000 seconds. The second scenario added TCP cross traffic from the five grey nodes shown in Figure 1 to compete with the TCP flow from the single node using the TCP variant under study.

Figure 2 shows simulation results for the first scenario. In this simulation the aggression factor for Stochastic TCP is set to Φ=1, which is equivalent to a set of parallel TCP streams using SACK streams. The standard deviation for each set of experimental results is shown as an error bar. In this figure, as well as in Figures 3 and 4, *Virtual Parallel TCP Streams* refers to the number of virtual TCP streams used in Stochastic TCP. For Φ=1, the behavior of a set of *N* virtual Stochastic TCP streams is similar to a set of *N* standard parallel TCP streams

Figure 3 shows the throughput of the TCP variant in competition with five cross-traffic streams. Each category is a stacked graph, with the light grey bar on the bottom representing the throughput of the TCP variant, and the dark grey bar on top showing the aggregate throughput of the five TCP cross-traffic streams. The standard deviation of the set of results for each experiment is shown as an error bar. The left half of the figure shows the throughput of a number of TCP variants, and the right half shows Stochastic TCP throughput (with aggression factor Φ=*n*) with an increasing number of virtual parallel TCP streams. Results are shown in this figure for the long router queue, with includes losses from queue overflows and the ns-2 statistical loss model; and for the long router queue of 1,000 packets, in which losses arise from the ns-2 loss model alone.

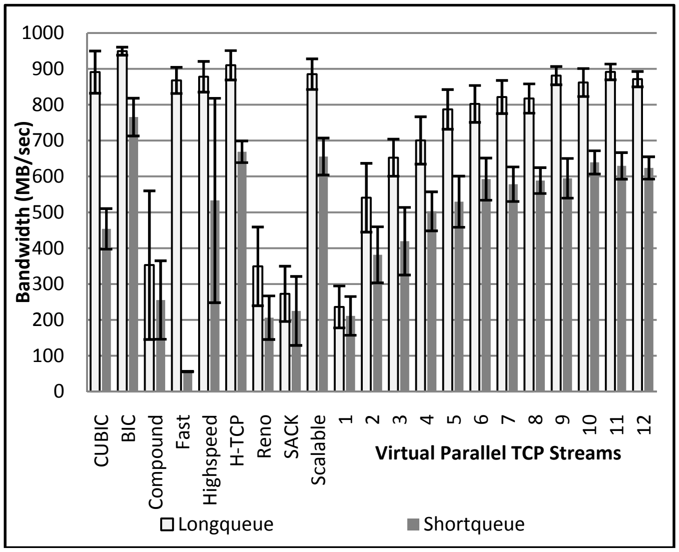

**Figure 2. TCP Throughput for TCP Variants and Stochastic TCP**

The TCP variants with the highest throughput are BIC, Scalable, and CUBIC.

Figure 4 shows simulation results in the short router queue simulation scenario, in which losses arise from the ns-2 loss model and router queue overflows. In this scenario, Scalable, BIC, and H-TCP provide the highest throughput.

### 6.1 Effectiveness vs. Fairness

The results shown in Figures 2 through 4 demonstrate that many TCP variants improve throughput compared with SACK and Reno. However, many do so at the expense of fairness – increasing throughput by stealing bandwidth from other flows is unfair, and may lead to congestion collapse if all users of a shared network begin an "arms race" of ever more aggressive TCP variants. In terms of fairness, Scalable-TCP was the worst, stealing significant amounts of bandwidth. The variants that are the fairest include Compound TCP, SACK and Reno, and Stochastic TCP. Stochastic TCP increased throughput by increasing

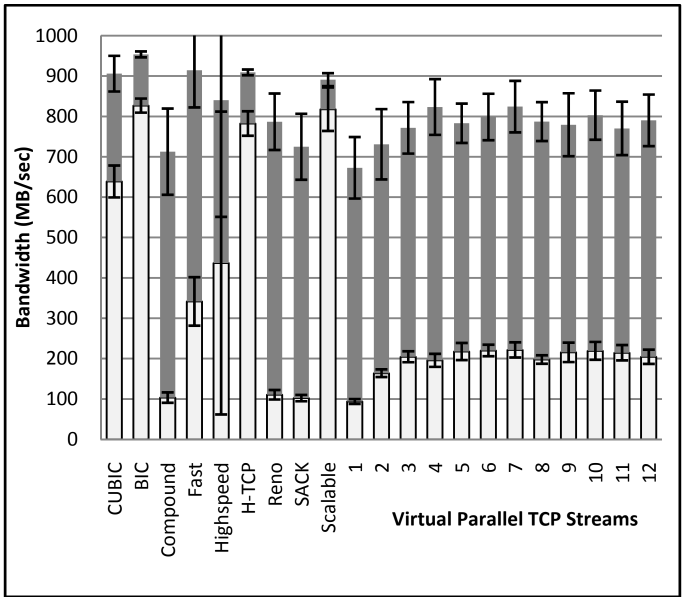

**Figure 3. Aggregate Throughput of TCP Variants Competing with 5 Cross-Traffic Streams, Long Router Queue. Dark bars in the stacked bar graph represent 5 Cross-Traffic Streams, and light bars represent TCP variant throughput.**

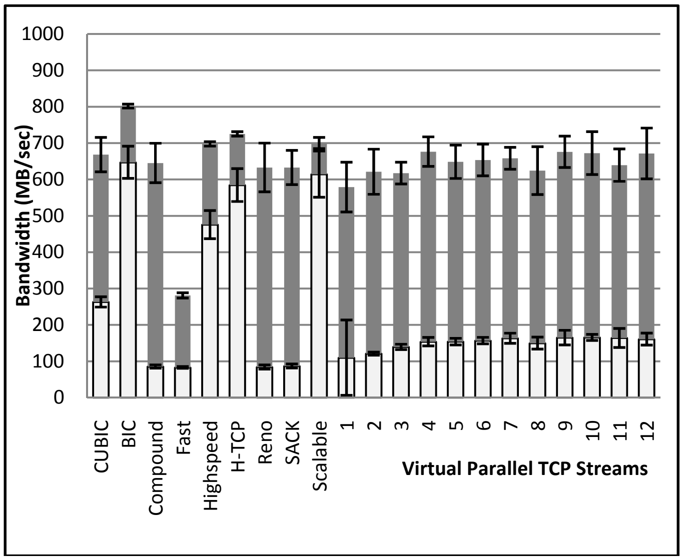

**Figure 4. Aggregate Throughput of TCP Variants Competing with 5 Cross-Traffic Streams, Short Router Queue. Dark bars in the stacked bar graph represent 5 Cross-Traffic Streams, and light bars represent TCP variant throughput.**

the number of virtual parallel TCP streams, but did so by utilizing unused bandwidth on the network path, rather than by stealing.

Based on simulation results, it is clear that using H-TCP, BIC, CUBIC, Highspeed, or Scalable TCP for Teragrid applications may help the application improve throughput, but only at the expense of stealing bandwidth from other applications sharing the network. Stochastic TCP could improve throughput compared with SACK and Reno, but only to the extent to which it would not steal bandwidth from competing TCP streams. Compound TCP, which is the new TCP variant built into Microsoft Windows Vista, proved to make an appropriate tradeoff between effectiveness and fairness, and thus would be a good TCP congestion avoidance algorithm for applications to use on the Teragrid network.

## 7 Related Work

ModelNet [12] is a large-scale network emulator that can be used with a clustered environment for evaluating distributed applications in a wide-area network environment. ModelNet provides a simple packet loss model based on a normal distribution. We chose to extend the ns-2 loss model, which provides a richer set of stochastic modeling functionalities to include a Weibull loss model.

Emulab [12] is a network testbed based on a cluster computing system that allows researchers to build and deploy kernels and applications within the context of a cluster computing environment. Emulab could be used for simulating the packet loss models and network simulation model developed for this paper. However, we found that BoilerGrid was adequate to meet our simulation needs. For future work, we are planning to investigate the use of Emulab.

## 8 Conclusions

The first step in building a network application testbed for developing and testing cyberinfrastructure components and systems is to create an accurate model of the network on which the cyberinfrastructure operates. In this paper, we described our efforts to build a network simulation testbed based on the ns-2 simulator that could accurately replicate the observed performance characteristics of the network that affects application performance. We assessed a number of high speed TCP variants, and measured the relative effectiveness and fairness of each variant under a number of conditions. We found that many aggressive TCP variants can improve throughput, but only by stealing bandwidth from competing flows. Based on simulation results, we determined that Compound TCP and Stochastic TCP can improve throughput compared with Reno and SACK, but will not do so by stealing bandwidth from competing TCP streams.

## References

1. *Ns-2 Network Simulator*. [cited; Available from: http://www.isi.edu/nsnam/ns.
2. *NISTNet network emulation package*. http://www.antd.nist.gov/itg/nistnet/ [cited.
3. Luigi, R., *Dummynet: a simple approach to the evaluation of network protocols.* SIGCOMM Comput. Commun. Rev., 1997. **27**(1): p. 31-41.
4. Xinjie, C. *Network simulations with OPNET*. in *Simulation Conference Proceedings, 1999 Winter*. 1999.
5. Hacker, T. and P. Smith. *Stochastic TCP: A Statistical Approach to Congestion Avoidance*. in *Sixth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet 2008)*. 2008. Manchester, UK.

6. Jonathan, S. and P. Craig, *When the CRC and TCP checksum disagree.* SIGCOMM Comput. Commun. Rev., 2000. **30**(4): p. 309-319.

7. Osterman, S. *tcptrace.* 2008 [cited; Available from: http://www.tcptrace.org.

8. *MathWave EasyFit.* 2008.

9. Ross, S.M., *Introduction to Probability Models.* 2000: Academic Press.

10. Smith, P., T. Hacker, and C. Song. *Implementing an Industrial-Strength Academic Cyberinfrastructure at Purdue University.* in *Second Workshop on Desktop Grids and Volunteer Computing*. 2008. Miami, FL.

11. Wei, D. and P. Cao, *NS-2 TCP-Linux: an NS-2 TCP implementation with congestion control algorithms from Linux*, in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*. 2006, ACM: Pisa, Italy.

12. Diwaker, G., et al., *To infinity and beyond: time-warped network emulation*, in *Proceedings of the 3rd conference on 3rd Symposium on Networked Systems Design \& Implementation - Volume 3*. 2006, USENIX Association: San Jose, CA.