Simulation of the TeraGrid Using SSFNet[1]

James A. Rome
Oak Ridge National Laboratory, Oak Ridge, TN 37831-6016
jar@ornl.gov

Abstract

TeraGrid is the NSF-sponsored high-speed 40 Gbps backbone that is an essential part of Grid computing. The 40 (and 30) Gbps links are made of bonded parallel 10 Gbps pipes. A hash is used to select the pipe for each connection. This simulation addresses the issue of how effectively the traffic gets split among the parallel pipes. In the simulation TeraGrid is connected to 33 servers and 66 clients. 1-GB ftp sessions are launched until packets are dropped. This drop occurs in the nodes with the most non-uniform hash to select the session pipes. Losses begin to occur when TeraGrid is at just over 50% of its rated capacity because the effective bit rate and current pipe utilization change as packets go from node to node.

1. Introduction

Simulation of computer networks is much cheaper, faster, and easier than actually building and testing the hardware itself. For example, TeraGrid [1] achieves its speed by bonding three or four 10 Gbps pipes to achieve greater throughput. We wanted to see whether this was an effective means of increasing the over-all throughput by performing a simulation. We used SSFNet 21], an open-source network simulation environment that can run under either Java or C++. The ORNL supercomputers consist of multiple nodes with multiple processors on each node, a configuration that is reminiscent of the network we were trying to simulate. Accordingly, with the help of Srdjan Petrovic (Dartmouth University), the Java version of SSFNet was ported to the IBM Eagle and Cheetah supercomputers at ORNL.

There is more to the simulation than just running the code. SSFNet uses Domain Modeling Language (DML) as its input to describe the network configuration. For a large network, with hundreds or thou-sands of elements, an automated way to read in network topology, plot, and manipulate it is required. Accordingly, a Java program NetViewer was developed to do these tasks. NetViewer is available at http://www.ornl.gov/~jar/NetViewer/Manual.htm.

The topology of a large network (e.g., ORNL's network) would be extremely tedious and error-prone to enter by hand. Even for the 700+ nodes of our TeraGrid simulation, it advantageous to use NetViewer to configure the individual nodes and to place traffic on the network. The topology of TeraGrid is shown in Fig. 1. The subnets at each site have been expanded to show the router, servers, and clients.

The unique feature of TeraGrid is the parallel bonded pipes that link the major routers. Four pipes are used between the Chicago and Los Angeles routers, and three pipes are used between these routers and the TeraGrid nodes at the San Diego Supercomputer Center (SDSC), Cal Tech, Argonne National Laboratory (ANL), and The National Center for Supercomputing Applications (NCSA) and the Pittsburgh Supercomputer Center (PSC).

2. SSFNet modifications and issues

SSFNet required several extensions in order to be able to simulate the TeraGrid. The built-in modules are limited to relatively low bandwidths because of a 64 kB buffer size. The buffers were all extended to be at least twice the delay-bandwidth product—hundreds of megabytes. Fortunately Java integers are long enough that the 1-GB transfer sessions did not cause integer overflows.

Major rewrites of the IP code were required in order to simulate the link-bonded parallel pipes that connect the major grid nodes. The BGP routing protocol selects one and only one path between a source and destination, so we had to change the code to allow parallel pipes to share the same ip addresses at their ends. Several different approaches were tried.

[1] The submitted manuscript has been authored by a contractor of the U.S. Government under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. This work was supported by the Office of Science, U.S. Department of Energy.

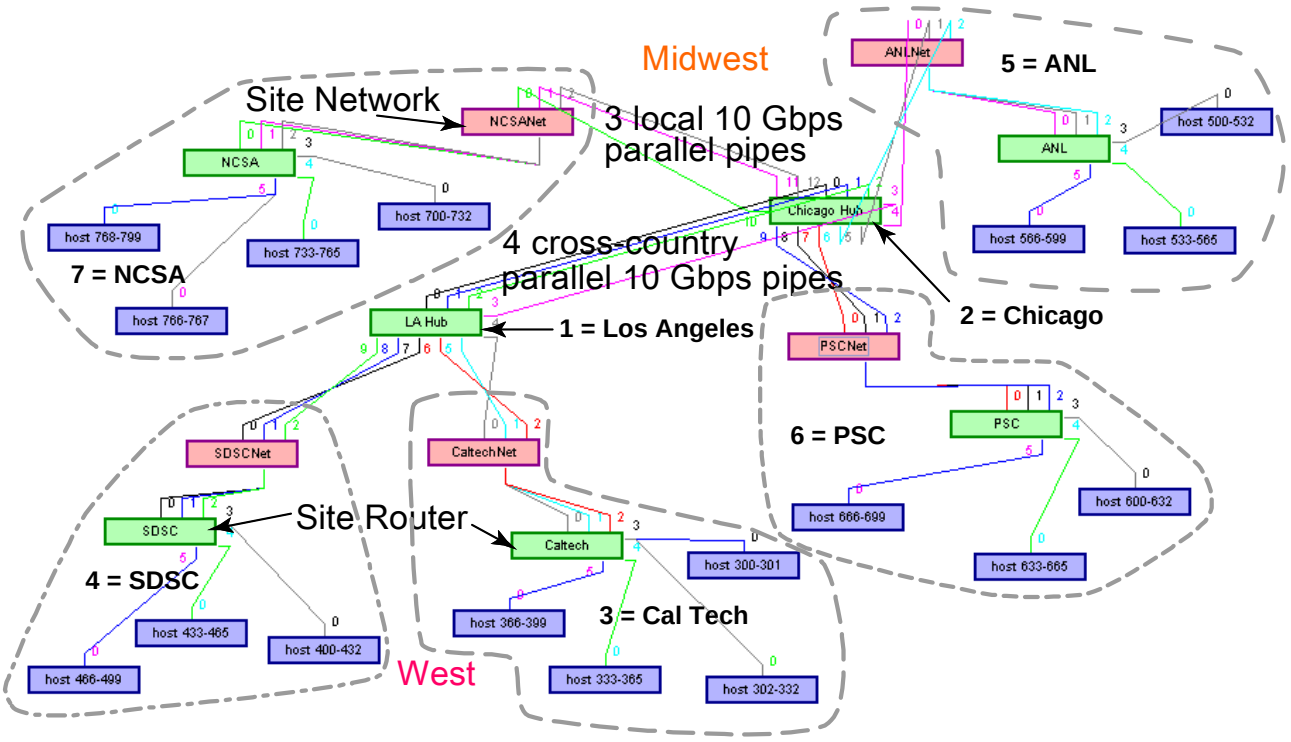

Fig. 1. The topology of TeraGrid. Routers are shown in green, site subnets in pink, and clients and servers in blue. The site subnets have been expanded to show the 33 servers and 67 clients attached to each site router.

The first method was to select a pipe randomly for each packet. It was soon discovered that out-of-order packets occurred and cause tcp retransmissions. To eliminate out-of-order packets, we decided to query the queue for each parallel pipe and place the packet on the shortest queue. Picking the lowest queue guarantees that the packet will arrive before the next packet and also enforce equal utilization of the multiple pipes. However, when the simulation was modified to select the lowest queue, the packets still eventually arrived out of order. After considerable effort, we discovered that the multithreaded nature of the simulation prevented success. In between the time the length of the queue was checked and the packet was placed onto the shortest queue, other threads had already put packets onto the queue so that it was no longer the shortest. The architecture of SSFNet prevented locking the necessary classes for exclusive use during this process.

We discussed the issue with engineers from Juniper Networks and learned that they use a hash to pick the pipe. The key for the hash is composed of the

- Source and destination IP addresses
- Source and destination ports
- Protocol (TCP only for now)
- Input interface number

The FCS hash from RFC 1662 Appendix C [3] is used. The resulting hash is divided modulo the number of pipes to select each pipe for the connection. But, the high bitrate data stream and low bitrate ACK stream are treated as equals. Because the source and destination are reversed, in general the two streams for a given connection will not use the same pipes.

In SSFNet, queues are simulated by determining the current queue delay, adding it to the link transit time, and calculating the time at which the packet will be delivered to the network interface card (NIC) at the other end of the link. Thus, the input and output queues are essentially one and the same. Initially the queue always fills up rapidly so that the server's round trip time (RTT) always includes the time to empty the queue buffer. As a result, if a server gets a second request from a client, it is difficult for the second stream to get started because the first stream has already filled up the buffer.

On a distributed supercomputer, the diagnostics must be confined to the code that runs on a given node of the supercomputer because files are local to each node until the job is completed. Accordingly, all the built-in SSFNet diagnostics were replaced in order to

determine the performance of the simulation and of TeraGrid. This locality of data makes it awkward to do things such as follow a single packet through the system. At the high bitrate in TeraGrid, gigabytes of data are collected for each router in just a few tenths of seconds of simulation time.

3. The simulation

75 simulated FTP sessions were used to create traffic on TeraGrid. Sessions were started randomly in the first 0.04 s of the simulation. Traffic was always directed from one site to another site, but not all sessions went across the main Los Angeles-Chicago link. One issue of performance is how well the hash works that determines pipe usage. The network architecture assumes that the law of large numbers will apply (i.e., many uncorrelated streams at once). However remembering that the TeraGrid is to be a computer backplane, the highly correlated 1 GB file transfers that we simulated are typical of the use it will see.

Figure 2 shows the pipe usage in the simulation. Because there are two streams for each connection, the sum of the bar heights are twice the number of connections, although not all connections go through each link. The hashes for the cross-country link have 4 bars, and the links from the LA and Chicago nodes have 3 bars. Note that the link from router 6 to router 2 (Chicago) has the most non-uniform hash.

When simulating on a distributed supercomputer, one easy thing to diagnose is the packet flow at individual network nodes. Figure 3 shows the packets leaving router 2 (Chicago) with the symbol style varying according to the destination ip address. The disconnected points represent retransmissions. These kill the TeraGrid tcp throughput even though TeraGrid is well below its supposed capacity.

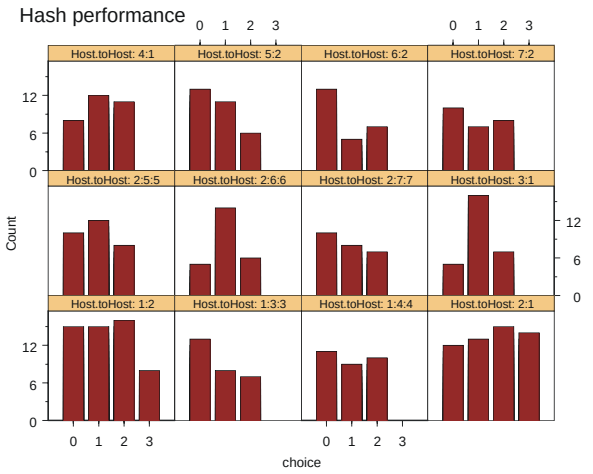

Fig. 2. The hash performance of the different parallel pipe links. The links with 4 bars represent the Los Angeles-Chicago link (both ways).

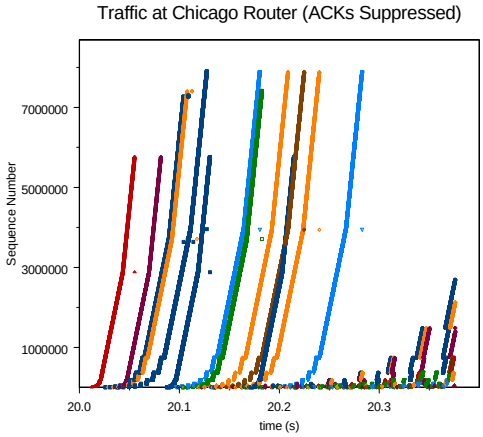

Fig. 4. Traffic leaving router 2 (Chicago). The detached points represent retransmitted packets.

4. Finding a needle in a haystack

Given the packet retransmissions of Fig. 4, the challenge is to explain why they occur. In other words, where did the packet get dropped? Following packets is very difficult in a distributed computer because the network nodes write to different files. Accordingly, we used a single computational node. With just one node, the code can write information for a given transmission to a separate file.

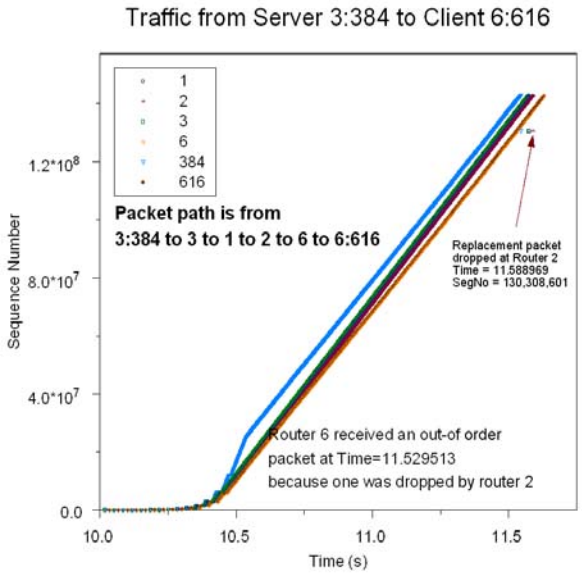

Fig. 5. Following packets from server 3:384 (Cal Tech) to client 6:616 (PSC). A packet gets dropped at the Chicago router (2) and re-transmitted. The retransmitted packet also gets dropped.

The results for one such transmission are shown in Fig. 5. The transmission delay is the width between the plots at any height. Initially, the output queue of the server fills up, and almost all of the delay is due to the time required to empty the buffer. However, as time increases, the delays occur at the router buffers until they get full and drop packets. The loss occurs between routers 2 and 6 (Chicago and PSC). The missing packet is shown at the client in Fig. 6.

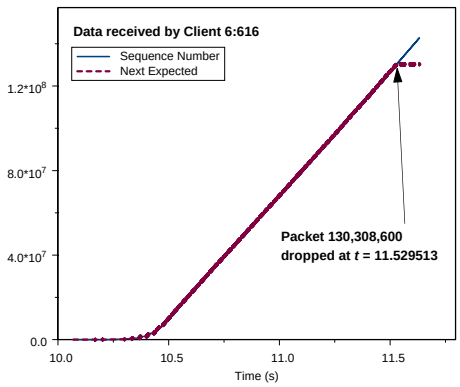

Fig. 6. The view of Fig. 5 at the Client showing the missing packet.

5. Discussion

The reason for the packet losses can be uncovered by plotting the performance of the Chicago router. Figure 7 shows the throughput (Mbps) as a function of the next hop destination. The traffic to router 6 (PSC) is just over 50% of its 30 Gbps capacity, while the traffic to router 1 (Los Angeles) is about 65% of its rated capacity.

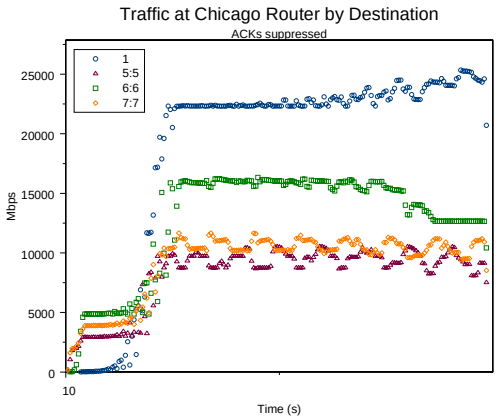

Fig. 6. The greatest traffic rate is between Chicago and Los Angeles (1), but still is only about 60% of the rated 40 Gbps throughput.

However, the links to the next hop are actually composed of bonded 10-Gbps pipes that are selected by the hashing algorithm. Figure 7 breaks up the flows of Fig. 6 by pipe. Corresponding to the hash performance of Fig. 2, we see that one pipe between routers 2 and 6 takes most of the traffic, exceeding the 10 Gbps capacity of the pipe, and leading to the ultimate packet drops.

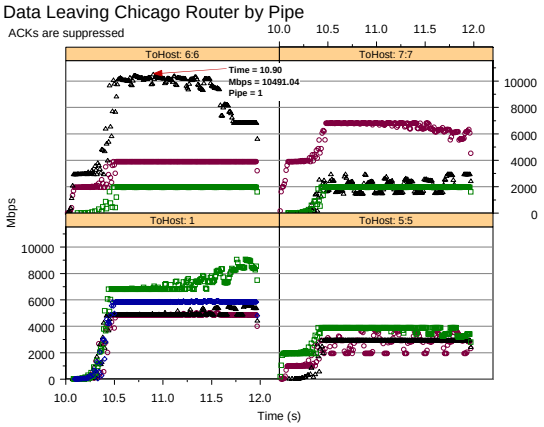

Fig. 7. Flows leaving the Chicago router by destination and pipe.

Compounding the problem is the large buffer size needed to accommodate the delay-bandwidth product.

The Server and Client have no knowledge of the bottleneck on their pipe between 2 and 6, which can suddenly be exacerbated by the presence of other connections that use this pipe. A large number of packets have already filled the output buffer of the Server, and are in the queue, so even if there was knowledge of the bottleneck, the Server rate could not be throttled back in time to prevent packet loss.

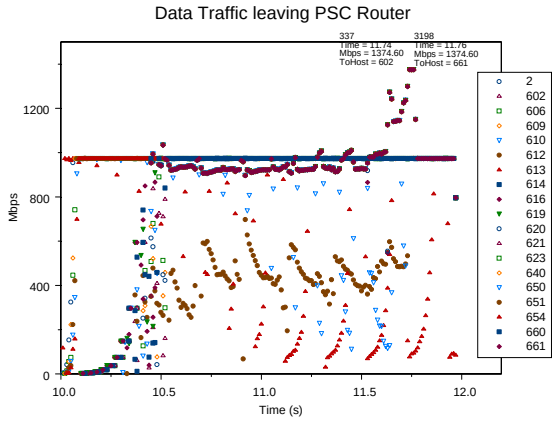

Fig. 8. Traffic leaving the PSC router in Mbps.

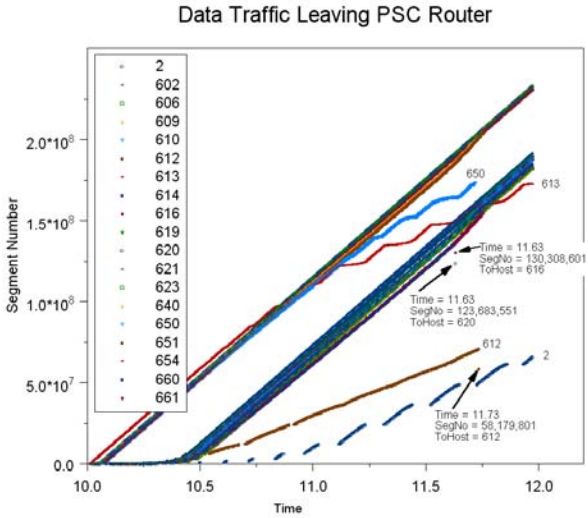

Fig. 9. Traffic leaving the PSC router by sequence number and flow. Retransmitted packets are shown.

Another problem arises due to the funneling of multiple pipes into a fewer number of pipes, or pipes with lower bandwidth. Although the server places packets onto TeraGrid at 1 Gbps, they can accumulate in the buffers of the network routers and be forwarded at a higher speed, ultimately arriving at the final router at a rate greater than the client can handle, at which point they will be dropped. This situation is shown in Fig. 8 where the packets are eventually sent to clients at a rate that exceeds the output NIC bit rate. The packets can be buffered for a time, but will be dropped when the buffer becomes full.

The same flows are plotted in Fig. 9 by sequence number, clearly showing the flows that have problems maintaining their bit rates, and flows that have problems getting started at all.

Conclusions

TeraGrid's bonded pipe architecture relies upon the law of large numbers to distribute the flows among the parallel pipes using a hash. Unfortunately, TeraGrid is supposed to carry large bulk data transfers rather than numerous small sessions. If the hash is non-uniform, individual pipes can exceed their capacity. In this simulation, problems arose at just over half of the rated bandwidth capacity. Changes in the available bandwidth going from node-to-node can cause packets to exceed the capacity of the next link. Accordingly, it is best if the clients and servers also have 10-Gbps network interfaces.

Acknowledgments

The author would like to thank his ORNL colleagues William R. Wing and Thomas H. Dunigan for their helpful advice. Srdjan Petrovic of Dartmouth University ported SSFNet to our supercomputers.

References

[1] http://www.teragrid.org/
[2] http://www.ssfnet.org/
[3] http://www.ietf.org/rfc/rfc1662.txt