SAND2008-0307C

Comparing Large File Transfer Methodologies on the 10-Gigabit ASC WAN

Jason S. Wertz
Network Design and Operations
Sandia National Laboratories
Albuquerque, NM
jswertz@sandia.gov

Lawrence F. Tolendino
Network Design and Operations (Retired)
Sandia National Laboratories
Albuquerque, NM
lftolen@sandia.gov

Don Gruenbacher
Electrical and Computer Engineering
Kansas State University
Manhattan, KS
grue@ksu.edu

Chris Lydick
Electrical and Computer Engineering
Kansas State University
Manhattan, KS
lydick@ksu.edu

John Sherrell
Electrical and Computer Engineering
Kansas State University
Manhattan, KS
jms7373@ksu.edu

ABSTRACT

The Advanced Simulation and Computing (ASC) program wide area network (WAN) connects the high performance computing (HPC) resources at the Department of Energy/National Nuclear Security Agency (DOE/NNSA) main weapons laboratories. The network allows the HPC resources to be shared and provides a means for moving terabytes of data between the sites. This paper evaluates different protocol approaches for the large data transfers in this unique environment by implementing simulation models. The different protocols evaluated for this network are: 1) TCP; 2) Parallel FTP (PFTP); and 3) SCTP. The performance of the TCP and PFTP approaches is compared via the modeling results and validated by laboratory tests. Results for the basic functionality of the SCTP approach are also provided. Proposed modifications, such as concurrent transfers using SCTP multihoming which should lead to enhanced performance, are also discussed.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Applications—*Parallel FTP, SCTP*

General Terms

Algorithms, Performance, Reliability, Verification

Keywords

Protocol Modeling

1. INTRODUCTION

The Advanced Simulation and Computing (ASC) program wide area network (WAN) connects together the high performance computing (HPC) resources at the Department of Energy/National Nuclear Security Agency (DOE/NNSA) main weapons laboratories. The HPC resources are critical elements in maintaining the US strategic nuclear stockpile since modeling and simulation have replaced testing. The network allows the HPC resources to be shared and provides a means for moving terabytes of data between the sites.

The interest in modeling and simulation of nuclear weapon performance was motivated by the Nuclear Test Ban Treaty and the desire to maintain a safe and effective nuclear weapon stockpile. The Advanced Simulation and Computing Initiative (ASCI) program, the precursor to the ASC program, was instituted to support and encourage the development of a modeling and simulation infrastructure and was focused on the supercomputers, visualization systems, and assorted support systems located primarily at the three main weapons laboratories: Los Alamos National Laboratory (LANL), Lawrence Livermore National Laboratory (LLNL), and Sandia National Laboratories (SNL). In recent years, the ASC program has continued this work providing the operational support and continual development required to utilize the implemented high performance computing environment.

Part of the original implementation strategy for these expensive hardware systems was to create the ASCI WAN for sharing precious ASCI resources throughout the weapons complex. The performance of the current ASC WAN strongly impacts the practicality of resource sharing and is a critical aspect of the program. As modeling and simulation play an increasingly important role in the maintenance of the nuclear stockpile, every effort must be made to provide users access to the ASC modeling resources regardless of the their location.

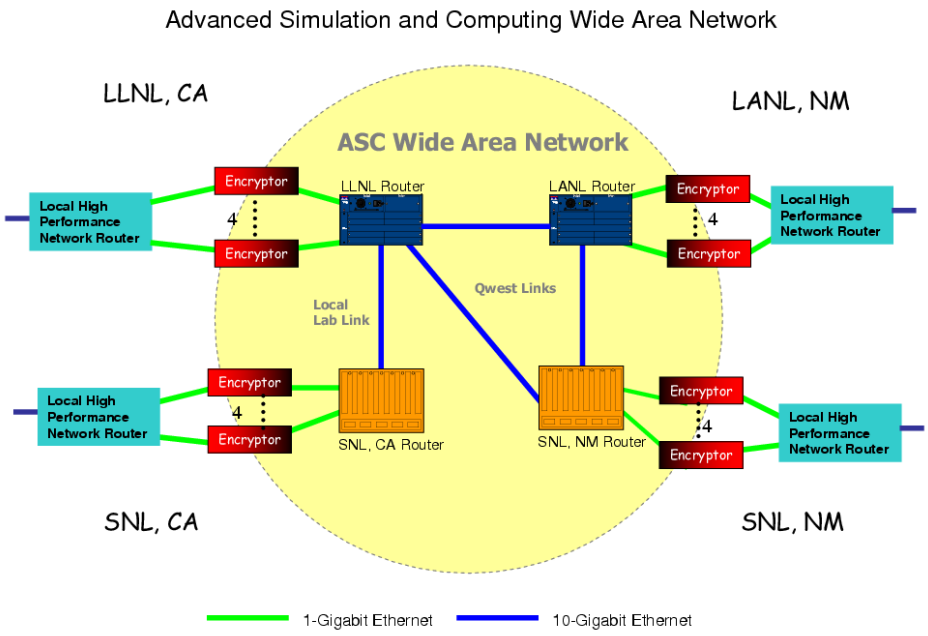

Figure 1: ASC Wide Area Network

1.1 Current ASC WAN

The current incarnation of the ASC WAN is based on 10 Gbit/s dedicated network links provided by Qwest Communications. These links are configured in a ring topology as shown in Figure 1.

Various technologies were used to create the network including Ethernet and the ubiquitous TCP/IP protocol set. Given the initial ASC performance goal of achieving a 200 Mbyte/s transfer rate for large data files, the ASC WAN engineering included parallel data streams in order to overcome the performance limitations of existing network technologies. Parallel data paths and multiple TCP/IP data streams are combined in the WAN to provide high performance and reliable data delivery over long distances. Such high performance comes at the cost of added complexity and increased maintenance as any design changes or trouble shooting activities are more difficult.

While the ASC WAN has successfully provided users with remote access to ASC resources for several years, the performance required to meet user needs has continually increased. This need for continual development and evolution of the WAN represents a challenge to the laboratories' staff because the WAN has taken on a critical production role in the ASC program. Making changes to the production network requires great care and planning if production operations are not to be impacted. Therefore, software modeling has been developed to analyze current network performance and predict the effect of proposed changes. The software modeling is augmented by a well-equipped data communications development laboratory where bench top WAN simulations can be exercised to evaluate various WAN traffic scenarios. The network development laboratory provides the software model validation necessary to build confidence in the software models.

The OPNET Modeler® software package was selected for modeling the ASC WAN and several custom model elements were developed to support the unique aspects of the WAN. These custom models include the implementation of the Stream Control Transmission Protocol (SCTP) and a model of the Parallel File Transfer Protocol (PFTP) data movement application. The remainder of this paper discusses the development of these custom models and their application in the network model of the ASC WAN as well as other enhancements that were made to the OPNET Modeler environment.

Each custom OPNET Modeler element served to increase the ASC WAN model fidelity and usefulness.

2. TOOLS

There were two main tool sets for this study. The first was the modeling and simulation software. The second was a hardware lab for verification and validation of the software results. Both pieces were critical to the success of the project. Modeling, using the OPNET Modeler software, allowed for many scenarios to be run in a short amount of time without consuming production networking resources and without consuming resources in the test lab. Once promising and realistic appearing scenarios were produced from the model, the important pieces were assembled in the lab and the software scenarios were reproduced and tested. The results from both the software tests and the hardware tests were then compared to determine if the software results were actually realistic and accurate.

2.1 OPNET Modeler

OPNET Modeler is a network modeling tool. It has a GUI front end to assist in visualizing a network and a 3-layer hierarchical structure. The lowest layer is the process model

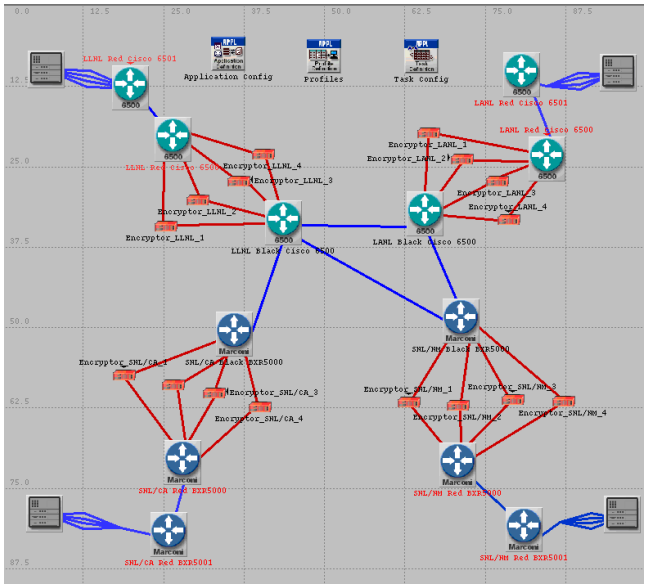

Figure 2: ASC Wide Area Network Model. Red links denote 1 Gigabit Ethernet. Blue links denote 10 Gigabit Ethernet.

layer which uses C++ to define each individual component or application as a state machine. The second layer is the node layer which combines process models into complete devices such as routers, servers, or firewalls. The third layer is the network layer which combines node layer devices into a network. This layer can be subdivided into multiple network layers. This is very useful for large complicated networks.

2.2 Lab Testing

The networking lab consisted of three main types of devices. The first type was the hosts, which generated and received traffic. The second type was the networking components, which provided routing and switching for the tests. The final piece was a delay simulator which generated the delays which simulated long distance communications within a lab.

There were four Linux hosts running Feisty Fawn Ubuntu (2.6.20 SMP kernel). Each host was powered by dual Opteron® 270 processors and had two gigabytes of system memory. Two of the hosts had fiber 1 Gigabit Ethernet interfaces and two of the hosts had 10 Gigabit Ethernet interfaces. All four hosts also had copper 1 Gigabit Ethernet ports that were used as control channels.

A couple of different routers were used for the tests. The first was a Foundry MG8 router and the seconds was a Foundry MLX-8 router. Both routers were equipped with 1 and 10 Gigabit Ethernet ports and could perform switching as well as routing functions. In general, traffic was routed in tests where a mixture of both 1 and 10 Gigabit Ethernet traffic was generated, while traffic was switched in tests using strictly 1 Gigabit Ethernet or strictly 10 Gigabit Ethernet.

The last major component of the hardware test lab was the Adtech AX/4000 box. Two 1 Gbit/s impairment modules were used for the tests. In general to emulate the ASC WAN, a delay of 40 ms was used. This is the average delay value on the production WAN. A few tests were run with other delay values ranging from 0 ms to 50 ms to test transfers sensitivity to delay.

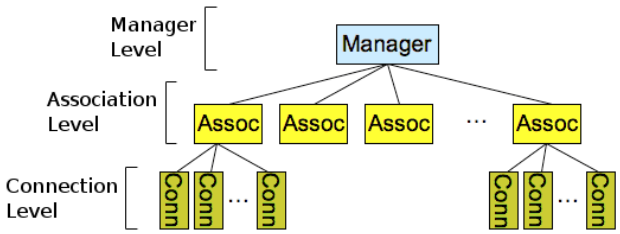

Figure 3: SCTP Model Hierarchy

3. MODELING

The following sections describe the development of Opnet models for SCTP and PFTP.

3.1 SCTP Model

Stream Control Transmission Protocol (SCTP) is a recently developed alternative transport layer protocol to TCP that provides various new capabilities that are attractive to the ASC WAN. Among its features is the ability to support multi-homed nodes to aid in redundancy and failure recovery. The SCTP standard [12] defines how connections across different interfaces are monitored to determine when a failure occurs so that the primary connection for data transfer will not reside on an interface with a failure in its path. When compared to TCP, this should be beneficial since the connection does not need to wait for a time-out to restart on an interface that likely has a bad path to the other endpoint.

There are many other differences between SCTP and TCP. Some of the key new features of SCTP are:

- SCTP supports multihoming,
- Streams are message-based instead of byte-based (TCP),
- SCTP establishes associations between computers, where each association can consist of multiple IP-pair connections,
- Association establishment involves a 4-way handshake that reduces the threat of Denial of Service (DoS) attacks, and
- SCTP segments are comprised of one or more chunks, where a chunk may contain data and/or control information.

3.1.1 Implementation

Because OPNET does not currently provide a model of SCTP, the authors chose to develop such a model for use in the ASC WAN simulation test bed. The OPNET model consists of three different layers (manager, association, and connection) as shown in Figure 3.

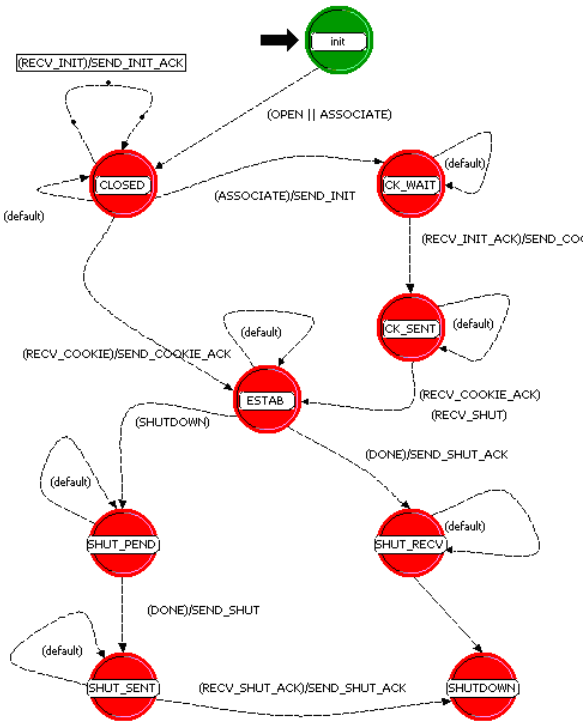

Figure 4: SCTP Association Process Model

In SCTP, a different connection can be established for each of the different interfaces available to reach a remote computer. A group of these connections being used to support a specific application data transfer is considered to be an association. The manager is responsible for directing all traffic to the corresponding associations.

The development of the OPNET SCTP model was based on adapting the TCP model. The manager process in TCP was duplicated and adapted to fit the needs of the SCTP manager process. Support for various packet formats and compatibility with adjacent protocol layers was developed. Both the association and connection process models in SCTP were adapted from the TCP connection process model. In the association process model, a new state diagram representing the SCTP association establishment 4-way handshaking was defined as shown in Figure 4. Likewise, the connection process model shown in Figure 5 indicates the role of heartbeats (HBs) in maintaining non-primary connections with the remote computer.

3.1.2 SCTP Model Validation and Verification

Testing and verification of the SCTP model is being accomplished through two independent approaches. The first compares the simulation output of a simple scenario of two computers connected via a switch to the actual traffic in the equivalent laboratory scenario. Traffic is generated using iperf in the physical configuration and a similar application in OPNET.

The second form of verification will be achieved by developing OPNET test configurations that compare to those given

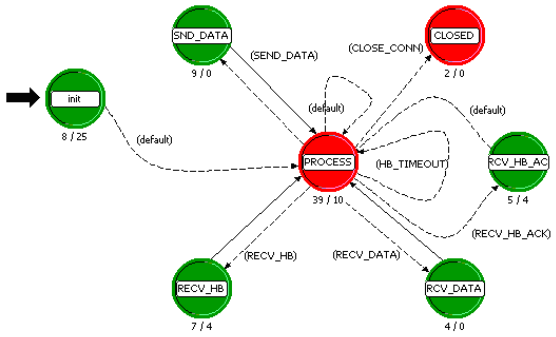

Figure 5: SCTP Connection Process Model

in [8, 9] and comparing the OPNET model results to those presented in [8, 9].

3.1.3 SCTP Model Enhancements

While the typical use for multihoming is to allow more robust networking options for wireless nodes that have more than one interface, the use of multihoming here creates distinct IP streams between two nodes. The primary benefit is that each IP stream can be processed through a different IP encryptor, thus overcoming any constraints that assignment to a single encryptor presents [3, 4, 7].

3.2 Parallel FTP

Parallel FTP (PFTP) is a client-server application designed to efficiently transfer large files[1] over high-speed, long-distance networks. The weapons laboratories helped develop PFTP as part of the High Performance Storage System (HPSS) collaboration and currently use the application to transfer scientific data sets over the ASC WAN [11]. Unlike common file transfer applications such as FTP and HTTP, which transfer a file serially over a single TCP stream, PFTP transfers many parts of a file concurrently by *striping* the file over many parallel TCP streams [2].

The OPNET model was developed in the application layer, and can be deployed on networks in the same way as the standard OPNET application models like FTP and email. Also like standard OPNET models, it provides a user-configurable attribute interface and output statistics that can be collected during simulations for later analysis. Users can configure striping and load balancing parameters and HPSS emulation (all discussed later), and file sizes of transfers. Output statistics include average file transfer rate, and aggregate throughput (in bytes or blocks) over time. The model also implements failure recovery, but this feature is untested and will not be considered further here, mainly because the failure recovery capabilities of real-world PFTP are undocumented. Other network parameters that have proven important in simulation results include TCP receive window sizes, TCP congestion control algorithms, end host processing speed, router queue sizes, and maximum transmission

[1]Normally on the order of gigabytes, although scalable at least up to terabytes [6]

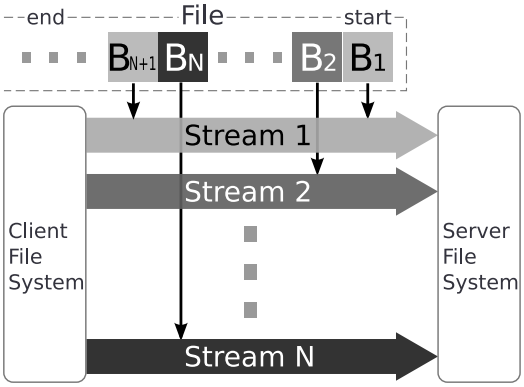

Figure 6: PFTP striping a file transfer over four parallel TCP streams.

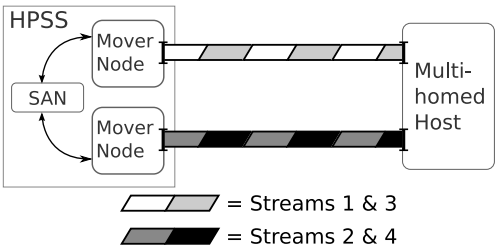

Figure 7: PFTP performs load balancing by opening TCP streams on multiple physical paths. Here, two paths carry two streams each.

unit (MTU) of network interfaces (the ASC WAN uses 9000 byte jumbo Ethernet frames).

The striping procedure used by both the PFTP model and the actual application [2] is illustrated in Figure 6. When a user initiates a parallel transfer, PFTP establishes N concurrent TCP connections between the client and server, where N is called the *stripe width* and is user-configurable. The file is divided into blocks of equal size, where the *block size* is also user-configurable. Now, the file can be thought of as a sequence of blocks, which we label $B_1, B_2, \ldots$, as shown at the top of Figure 6. Moving sequentially through the file, blocks are assigned round-robin to TCP streams, so that the ith stream is assigned blocks $B_i, B_{i+N}, B_{i+2N}, \ldots$, and transfers them in that order [2]. Put another way, files are striped at block granularity over N parallel TCP streams.

It is well known that network paths with a high bandwidth-delay product (BDP), like those on the ASC WAN, are often better utilized by an aggregate of parallel TCP streams than by a single TCP stream [1, 2, 5, 10]. Additionally, parallel streams enable a single PFTP transfer to utilize multiple physical paths, which allows PFTP to implement application-specific load-balancing, for example over four encryptor pairs between end hosts on the ASC WAN.

When configured to utilize multiple paths between the source and sink file systems, PFTP implements load balancing by distributing TCP streams round-robin over the paths [2]. Figure 7 shows two paths carrying two streams each. In both the model and the real application, streams on a given path are interleaved randomly (unlike the consistent pattern in the figure) because each stream is controlled by an independent PFTP child process. In OPNET, application models can only specify a destination host name, and not a specific IP address, when opening connections. To support load balancing in the PFTP model, it was necessary to modify the standard `tpal` OPNET model, which interfaces applications with the transport layer, to accept a destination IP address with application connection requests. On HPSS systems, the endpoint of each path is a separate "mover" node, and PFTP actually coordinates processes across nodes. The model does not support multinode configurations, but it does support simpler case of multihoming, where a single host has multiple interfaces connected to the network. Both cases are shown in Figure 7.

PFTP was developed as an interface to the HPSS parallel storage system. One crucial shortcoming of the current PFTP model is that it mostly ignores the impact of file system I/O on overall application performance. In reality, the network performance of PFTP can be limited by file system performance. One important characteristic of HPSS is that files move fluidly through a storage hierarchy, between slow, high-capacity tape media and faster cache storage levels. When a file is requested, it might be located anywhere in this hierarchy, and the location can have a profound effect on the latency encountered when accessing the file. The current model essentially simulates an ideal file system, with zero latency and infinite bandwidth. Another major aspect of HPSS is that it stripes logical volumes over physical storage media in a way analogous to our description earlier of PFTP striping a logical data stream over network sockets. Dedicated machines called *movers* access different storage media concurrently, and also host the child PFTP processes that serve as the transfer endpoints, as shown in Figure 7. Thus, optimal values of the stripe width and block size parameters depend on the corresponding parameters on the HPSS system. For example, if the PFTP stripe width is set smaller than the storage stripe width, then movers must at least partially aggregate and serialize physical storage before transferring over the network. The current model has no concept of storage parallelism, and does not model any efficiencies from matching network and storage parameters.

There are, however, other HPSS behaviors that the model is capable of emulating. HPSS forces PFTP to transfer blocks in a lock step, meaning that each stream must wait for an application-layer acknowledgment after transferring each block before beginning the next block. This can have a substantial impact on performance on high-latency networks since each stream sits idle for one RTT after each block transfer [11]. PFTP implementations have been built on non-HPSS platforms that do not require the lock step, and the model therefore allows users to enable or disable the lock step in simulations. Finally, the model can simulate checkpoints, where extremely large transfers on HPSS systems are split into a sequence of smaller transfers. This allows failed transfers to resume from the previous checkpoint, rather than start over from scratch. Tape machin-

ery failure is particularly problematic because it can take quite a long time to recover, so checkpoints are important for HPSS. However, they can harm performance because they force streams to synchronize at checkpoints, which can cause idle waiting time for some streams. The checkpoint interval is a configurable parameter in th model.

TCP streams each independently maintain a congestion window (*cwnd*), which limits at any given time the number of bytes of sent data for which the sender has not received acknowledgment from the receiver. TCP employs a congestion control algorithm where senders increase *cwnd* as the sequence of acknowledged segments advances, and decrease *cwnd* when packet loss is detected. In this way, TCP streams discover the bandwidth available to them while attempting to fairly share bandwidth with other streams. Additionally, receivers advertise their available buffer capacity to the sender as the receive window (*rwnd*), which becomes the limiting factor when it is smaller than *cwnd*. Thus, the maximum bandwidth that a TCP stream can utilize at any given time is

$$BW_{max} = \frac{\min(cwnd, rwnd)}{RTT}, \qquad (1)$$

where RTT is the round-trip time seen by the sender between successfully transmitting a segment and receiving the corresponding acknowledgment. For a stream to fully utilize BW_{neck}, the bottleneck bandwidth of the end-to-end network path carrying the stream, both *cwnd* and *rwnd* must be larger than the BDP ($RTT \cdot BW_{neck}$) of the path.

We now discuss a hypothetical PFTP transfer on the ASC WAN between Albuquerque, NM and Livermore, CA. PFTP sees four paths between the hosts, each defined by a client interface paired with a server interface, and each with $RTT =$ 40 ms and $BW_{neck} = 1$ Gbit/s at the encryptors. If the stripe width is set to 4, PFTP will open a single TCP stream on each path, and each stream will see a BDP of (1 Gbit/s $\times$ 40 ms) = 5 MB. The streams begin in the *slow-start* phase of the TCP congestion control algorithm: *cwnd* starts at twice the maximum segment size ($2MSS$) and is incremented by MSS for every acknowledged segment until the sender detects packet loss. Thus, *cwnd* grows exponentially with time and is fully "open" after $\log_2(BDP/2MSS)$ round trip times, or about 0.325 s for our transfer with $MSS \approx 9$ kB. However, slow-start traffic is "bursty", which can load router queues that funnel data from faster interfaces to slower interfaces, leading to packet loss. The bottleneck 1 Gbit/s interfaces on the ASC WAN have queues that are an order of magnitude larger than the end-to-end BDP, so we don't expect packet loss unless the window is not limited by *rwnd*, and *cwnd* continues to grow exponentially after BW_{neck} is fully utilized.

3.2.1 Validation and Verification

PFTP is proprietary and in general is deployed only on specialized computer systems. Tests are routinely run on the ASC WAN to obtain average PFTP transfer rate, but more detailed test data such as packet losses or time-varying throughput is difficult to obtain for validation purposes. However, the average transfer rates are useful measurements to validate against.

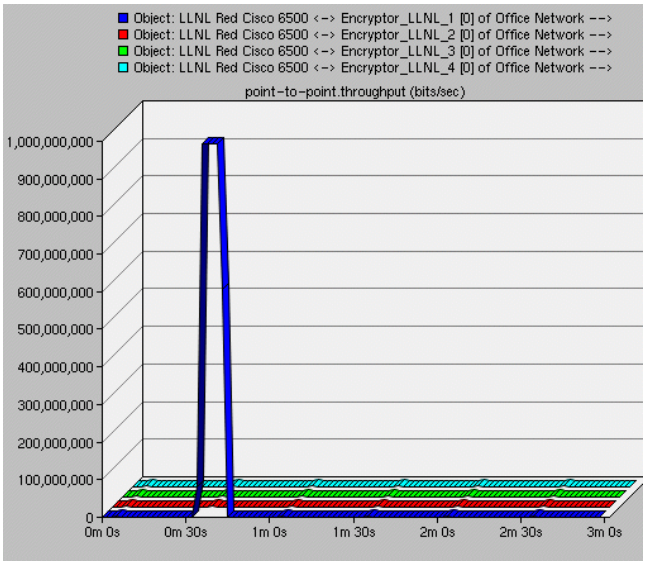

Figure 8: SCTP Traffic with Heartbeats on the ASC WAN

An analytical model for the aggregate bandwidth of parallel TCP streams developed and already validated by [5].

$$BW_{agg} \leq \frac{MSS}{RTT}\left[\frac{1}{\sqrt{p_1}} + \frac{1}{\sqrt{p_2}} + \cdots + \frac{1}{\sqrt{p_n}}\right] \qquad (2)$$

where p_i is the packet loss rate for stream i, assumed to be under 1/100.

4. RESULTS

The results below were obtained from each of the simulation models developed as part of this effort. The SCTP model and results are still being developed and evaluated. The PFTP model is more mature and subject to only minor adjustments.

4.1 SCTP

The SCTP model is currently able to show most of the general functionality using simple tests of the ASC WAN. Fine tuning and validation of the SCTP model is currently in the early stages. Figure 8 shows the behavior of a traditional SCTP transfer using four connections across one of the encryptor banks. One of the connections is designated as primary and carries all of the data traffic under normal circumstances. The other three connections consist of only heartbeat traffic (evident as small ripples) in order to maintain the health status of those connections.

Figure 9 shows the behavior of a similar transfer where the primary connection encounters a problem and the SCTP Association level assigns a different connection the primary designation.

Figure 10 shows a more balanced approach by implementing a load-balancing feature. These results show how SCTP can attempt to distribute the data traffic evenly across all active connections.

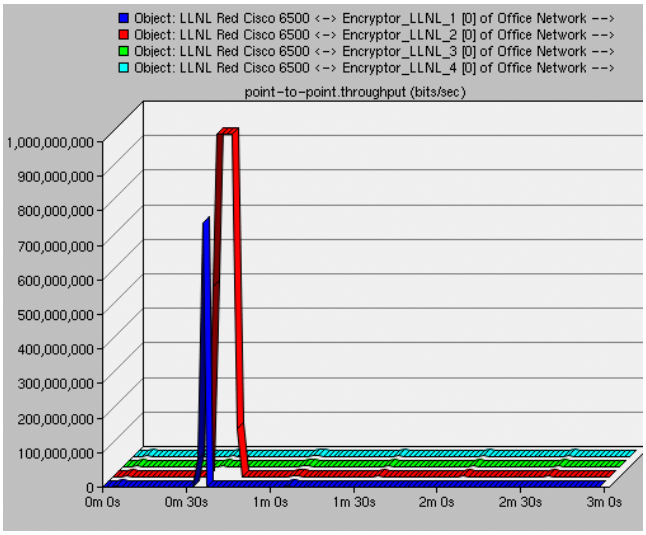

Figure 9: SCTP Automatic Failue Recovery on the ASC WAN

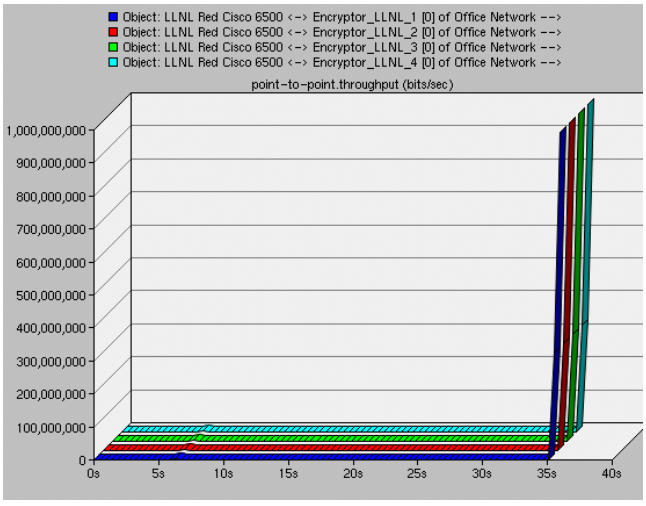

Figure 10: SCTP Balanced Traffic on the ASC WAN

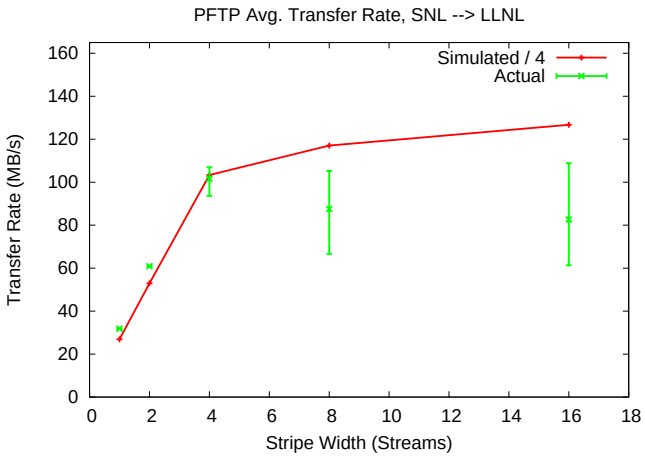

Figure 11: Simulated and actual average PFTP transfer rates for various stripe width values. The simulated values are scaled by 1/4.

4.2 PFTP

Figure 11 compares simulated transfer results to actual tests run on the ASC WAN. The graph plots the average transfer rate of 4 GB file transfers as stripe width increases from 1 to 16 streams. The simulated curve is scaled by 1/4. The simulation was performed in OPNET 14.0 with the PFTP application process models running on the ASC WAN network model. Both the test and the simulated transfer were between SNL in Albuquerque, NM and LLNL in Livermore, CA. Each transfer used 16 MB blocks and TCP buffer sizes set to 8 MB. Each test was run three times, and the bars on the plot show the minimum, maximum, and mean value for each data point. Router queues were set large enough on the simulation to eliminate packet loss. The simulated values were actually 4 times greater than on the plot, so they begin at 94 MB with only one stream and peak at 443 MB with 16 streams. The lack of accurate modeling of the file system can likely account for at least part of the discrepency. The tests were run with files that were in cache (no "tape load") in order to minimize the performance effects of the file system. The scaled version of the simulated curve is shown to highlight the correlation between the simulated and actual values as stripe width increases. It appears that the model is incorrect mainly by a simple scaling factor.

5. FUTURE WORK

The ASC modeling project is far from complete. Besides finishing the validation and verification of the SCTP model, several new modeling efforts are planned in the upcoming year. Testing will be performed with 10 Gbit/s delay simulators, a relatively new application called MPSCP will be tested, and validation and verification will occur not only in a lab, but on the production WAN itself. In future years, the modeling will be extended to include full disk-to-disk models and to help with planning for a petascale computing environment.

5.1 10 Gigabit Impairment

The lab protocol testing has only used a 1 Gbit/s impairment module to simulate network delays because, until re-

cently, a 10 Gbit/s impairment device wasn't available. A network emulator, the ANUE GE10 with 10 Gbit/s delay capabilities, was recently added to the testing lab. Future protocol testing will incorporate the new testing device to provide a more accurate lab emulation of the ASC WAN to compare with models.

5.2 MPSCP

MPSCP is an open source parallel file transfer application developed at Sandia National Laboratories approximately two years ago. It uses an encrypted channel to set up the connection (SCP), but transfers data over a user defined number of unencrypted paths and/or connections. This protocol is viewed as being promising for the ASC WAN for a couple of reasons. The first is that in simple trials on the WAN, MPSCP has shown itself to be approximately 12 times as fast as PFTP for transferring large numbers of small files . The test involved transferring five hundred 100 MB files and twenty-five 1.6 GB files, adding up to approximately 92 GB of data. PFTP completed all transfers in 5860 seconds while MPSCP took only 483 seconds. For large files (tens of gigabytes), MPSCP and PFTP appear to run at similar transfer rates. A second reason is the MPSCP is generally easier to configure than PFTP. This is because MPSCP doesn't require large, specialized configuration files on each node. It also uses a syntax that is very similar to standard FTP, which is well known amongst the high-performance computing community that these programs are targeted at.

5.3 Validation and Verification on the ASC WAN

Another planned step for the coming year is to move validation and verification (V & V) from primarily being done in the lab to being done primarily on the ASC WAN. This should produce a better benchmark for the software models to be compared to since the lab testing tends to produce a somewhat idealized system with fixed jitter and zero competing traffic. The one major drawback of testing on the live WAN versus laboratory test is that it is very easy to impair the production traffic with test traffic, so extra planning and coordination will be required before this testing is done.

5.4 Disk-to-Disk Models

The current models essentially simulate memory-to-memory transfers. In order to more fully match the whole system, the models will need to be extended to simulate disk-to-disk and disk-to-tape operations. This will be a considerable effort as there are many different disk and tape systems used in the ASC end hosts and backup nodes, but will add an important layer of fidelity to the models.

5.5 Petascale computing

Many resources are currently being directed at petascale computing at the ASC laboratories. Since it is economically infeasible to have a petascale computer located at each site, the ASC WAN will be required to connect personnel at the various sites to the petascale compute resources. The ASC WAN modeling effort can be leveraged to provide a very economical way to study the effects of the additional traffic loads on the WAN.

6. CONCLUSIONS

A model has been developed in OPNET to help reduce the cost of design, evaluatio n, and testing of the ASC WAN. The newest model includes the incorporation of p rocess models for both PFTP and SCTP. The performance of the PFTP model was va lidated against PFTP data obtained from the production ASC WAN. For small numbe rs of parallels streams (4 or less), the model compared very well against the pr oduction data. For larger numbers of streams, the match wasn't as good, but was still promising. It is believed that incorporating a model of the file system on one or both ends will improve the fidelity of the PFTP model. The SCTP model also shows promise, but it isn't ready for a full comparison versus either the production WAN or the test lab. These models will continue to be adjusted to be tter represent the actual traffic over the network.

7. REFERENCES

[1] M. Allman, H. Kruse, and S. Ostermann. An application-level solution to tcp's satellite inefficiencies. In *First International Workshop on Satellite-based Information Services (WOSBIS)*, November 1997.

[2] M. Barnaby. Parallel ftp: Realizing the maximum potential network bandwidth via multiple sockets. Technical Report SAND2001-1290A, Sandia National Laboratories, Albuquerque, NM, 2001.

[3] A. Caro, J. Iyengar, P. Amer, G. Heinz, and R. Stewart. Using sctp multihoming for fault tolerance and load balancing. In *Proceedings of ACM SIGCOMM*, 2002.

[4] C. Casetti, R. Greco, and G. Galante. Load balancing over multipaths using bandwidth-aware course scheduling. In *Proceedings of the 7th International Symposium on Wireless Personal Multimedia Communications*, 2004.

[5] T. J. Hacker, B. D. Athey, and B. Noble. The end-to-end performance effects of parallel tcp sockets on a lossy wide-area network. In *International Parallel and Distributed Processing Symposium*, pages 434–443. IEEE, 2002.

[6] The HPSS Collaboration. *HPSS User's Guide*, release 5.1, revision 1 edition, December 2003. Chapter 3: Parallel File Transfer Protocol.

[7] J. Iyengar, K. Shah, P. Amer, and R. Stewart. Concurrent multipath transfer using sctp multihoming. In *Proceedings of SPECTS 2004*, July 2004.

[8] A. Jungmaier, E. Rathgeb, M. Schopp, and M. Tuxen. Sctp: A multi-link end-to-end protocol for ip-based networks. *Int. J. electron. Commun. (AEU)*, pages 46–54, 2001.

[9] A. Jungmaier, M. Schopp, and M. Tuxen. Performance evaluation of the stream control transmission protocol. In *Proceedings of the 3rd International Conference on ATM*, June 2000.

[10] T. Kelly. Scalable tcp: Improving performance in highspeed wide area networks. *Computer Communication Review*, April 2003.

[11] J. S. King. Parallel ftp performance in a high-bandwidth, high-latency wan. Technical Report UCRL-MI-142491, Lawrence Livermore National

Laboratory, Livermore, CA, November 2000.

[12] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, and M. Kalla. Stream control transmission protocol. RFC 2960, Internet Engineering Task Force, Oct. 2000.