Automatic Buffer Tuning and Improved Security for High-Energy Nuclear Physics Data Transfers (DRAFT #3)

DOE Program Notice: DE-FG01-04ER04-01

Wu-chun Feng (PI), **Mark K. Gardner** (co-PI)
Los Alamos National Laboratory
P.O. Box 1663, M.S. D451, Los Alamos, NW 87545
Tel: 505-665-{2730,4953}, Fax: 505-665-4934, {feng,mkg}@lanl.gov

Roger Leslie A. Cottrell (PI), **Andrew B. Hanushevsky** (co-PI)
Stanford Linear Accelerator Center
2575 Sand Hill Road, Mail Stop 97, Stanford, California 94309
Tel: 650-926-{2523, 2063}, Fax: 650-926-3329, {cottrell, abh}@stanford.edu

May 3, 2004

Abstract

Transferring scientific data is one of the most important functions performed by DOE networks. Recently, as much as 76% of the traffic on the Abilene network was found to support data transfers. In order to improve network performance in support of the DOE scientific mission, we propose to add support for the dynamic right-sizing of TCP send and receive buffers to `bbcp` and `bbftp`, two applications widely used in the high-energy nuclear physics community. While we are at it, we will investigate adding authentication and authorization certificate support to improve the security of data transfers.

The DRS algorithm was designed to automatically and continuously adapt TCP buffer sizes to deliver high performance for scientific applications without requiring tedious hand tuning. The kernel implementation of DRS has been shown to improve transfer rates by a decimal order of magnitude over untuned TCP. However, the kernel implementation requires recompiling the operating system kernel with DRS support. Therefore, a technique for implementing DRS in user space has also been developed. It has been shown to improve transfer rates by 5x–8x for FTP transfers *without requiring kernel modifications*. We will therefore implement user-space DRS in the target applications.

Our approach has two phases. The first is to directly implement the DRS algorithm in `bbcp` or `bbftp`. The second is to develop a library, called `libdrs`, that makes supporting DRS much easier. The former ensures that the performance improvements of DRS are delivered into the hands of scientists as quickly as possible (i.e. within the first year). The latter reduces the effort required to add DRS support to arbitrary applications and speed deployments in all data intensive domains.

As part of the technology transfer, we will deliver production-ready versions of `bbcp`, `bbftp` and `libdrs` by the conclusion of the project. We will also demonstrate the performance of the user-space implementation of DRS in production and high performance testbeds networks with real HENP data. Because of the popularity of `bbcp` and `bbftp` and because we already have an active presence in the community, we will be able to achieve wide-spread deployment and have a significant positive impact on scientific discourse.

Contents

1 Background and Motivation

Transferring scientific data is one of the most important functions of DOE networks. For the week ending 2004/03/08, data transfers (defined as transfers over 10 MB) accounted for 25.7% of the 410.0 TB of traffic that traversed the Abilene network [1]. Furthermore, it appears that much of unidentified traffic (24.9%) also pertains to data transfers[1]. Therefore, it is likely that data transfers account for at least half of the traffic on Abilene.

In reality, however, the amount of traffic supporting data transfers may be over 75%. In order to achieve good wide-area network (WAN) performance the size of the TCP buffers for a connection must be large enough to ensure that the network can be kept full. The appropriate TCP buffer size is equal to the product of the bandwidth of the bottleneck link and the RTT of the connection, i.e. the effective bandwidth-delay product of the connection. From [1], the total traffic for Iperf packets (a tool used to measure achievable throughput for data transfers) and ICMP (to measure round-trip times) was 25.7%. Thus, up to 76.3% of the traffic for the week of 2004/03/08 pertained to data transfers.

As stated previously, TCP buffer sizes must be tuned appropriately in order to achieve high transfer rates. Indeed, many applications commonly used by scientists for transferring large quantities of data, such as `bbftp`, `bbcp` and GridFTP, have provisions for manually setting the buffer sizes. Others, such as `ftp`, do not have such capabilities built in and hence must rely on appropriately set defaults in the operating system (OS) [2, 3]. In all cases, buffer tuning requires determining the available bandwidth and the round-trip time.

In an attempt to automate the buffer tuning process, some applications actively probe the network before a data transfer commences to estimate the bandwidth-delay product [4]. Whether by hand or within an application, active probing pollutes the network with extra packets and hinders the transfer of real data. Even if the traffic needed to probe for network conditions is small, it becomes significant when many applications probe. As the data from 2004/03/08 shows, the amount of pollution can be as high as half of the amount of scientific data transfered. Clearly, a means of inferring the appropriate buffer sizes without polluting the network is needed.[2]

An orthogonal issue is the frequency with which the buffers are tuned. One-time tuning, automatically or by hand, assumes that the bandwidth-delay product is constant, or at least is reasonably stable. However, this is not always the case. Figure 1 presents the bandwidth-delay product between Los Alamos and New York at 20-second intervals. The bottleneck bandwidth ranges from 0.026 Mbps to 28.5 Mbps, with an average bandwidth of 17.2 Mbps. The RTT delay also varies over a wide range (119–475 ms) with an average delay of 157 ms. As a result, the bandwidth-delay product of the monitored connection can vary by 61 Mbit

[1] As of 2004/03/08, `bbcp` traffic is not yet recognized on Abilene. However, the amount of `bbcp` traffic observed at the Stanford Linear Accelerator Center over the course of a few days was similar to the amount of `bbftp` traffic measured on Abilene. Hence much of the unidentified traffic on Abilene is likely `bbcp` data transfers.

[2] It is sometimes suggested that the send and receive buffers can be statically set to an enormously large value so that tuning need never take place. There are several problems with this. First, TCP is susceptible to self-induced congestion when buffers are sized larger than the bandwidth-delay product of the network. This causes periodic packet loss and extremely long recovery times on networks with large bandwidth-delay products. The result is poor throughput. Next, allocating large buffers may place limits on the number of connections that can be established. Finally, allocating large amounts of buffer space can exhaust kernel memory on most operating systems. Exhausting kernel memory typically results in a system crash.

(nearly 8 MB), even over short intervals of time. A one-time estimate of the bandwidth-delay product is likely to (1) under-allocating memory and under-utilizing the network or (2) over-allocating memory and wasting precious oeprating system resources. (Unless the bandwidth-delay product is constant, a one-time choice will not be ideal throughout the lifetime of a connection.) Therefore, the bandwidth-delay product need to be dynamically inferred in order to ensure that data transfers proceed at full speed without wasting valuable system resources.

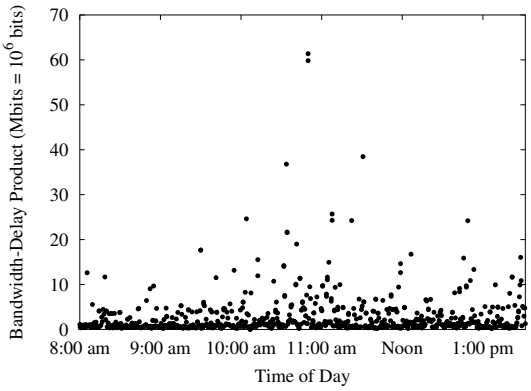


Figure 1: Bandwidth-Delay Product at 20-Second Intervals

We have developed an OS kernel-based technique, called Dynamic Right-Sizing (DRS), that passively infers the available bandwidth and round-trip time without injecting extra traffic into the network [5–7]. Because it infers the bandwidth-delay product through out the lifetime of the connection, it can respond appropriately to changing network conditions, such as those shown in Figure 1. As a result, the throughput between end hosts will only be constrained by the available bandwidth of the network and the efficiency of TCP's congestions avoidance, rather than inappropriately sized buffers.

Presently, the only other automatic buffer tuning technique is auto-tuning [8]. Auto-tuning implements sender-based buffer adaptation by fairly sharing buffer space on the sender. (It assumes that buffer space appropriately tuned on the receiver.) DRS, on the other hand, implements receiver-based buffer adaptation by sizing the receive buffers according to network conditions and available memory on the receiver. (DRS makes no assumptions about the buffer space on the sender. A sender without sufficient buffer space is expected to transmit at a slower rate.) Because it reduces the amount of buffer space advertized by the receiver to take into account available memory, the DRS prevents the sender from overrunning the receiver. In order to prevent buffer overrun on the receive with auto-tuning, care must be taken to ensure that sufficient buffer space is available on the receiver.

The installation of DRS in an OS kernel requires recompiling the kernel and administrative privileges to install it. Further it requires access to the kernel sources so DRS is not available for proprietary operating systems such as Solaris. Also, distributing DRS as a kernel patch is problematic as new patches need to be generated for each kernel version or users need to have sufficient expertise to fix problems when patches do not apply cleanly. Thus, DRS functionality is generally not accessible to the typical scientist.

Although, we anticipate that DRS will eventually be incorporated into operating systems such that its installation and operation are transparent to the end user, we have developed a portable user-space

implementation technique in the meantime. One user-space implementation incorporates DRS into FTP (drsFTP) [9]. Another is a prototype implementation of DRS in GridFTP [10]. Even though a user-space implementation does not have access to the state of a TCP connection, the throughput improvements (discussed in the next section) are still dramatic.

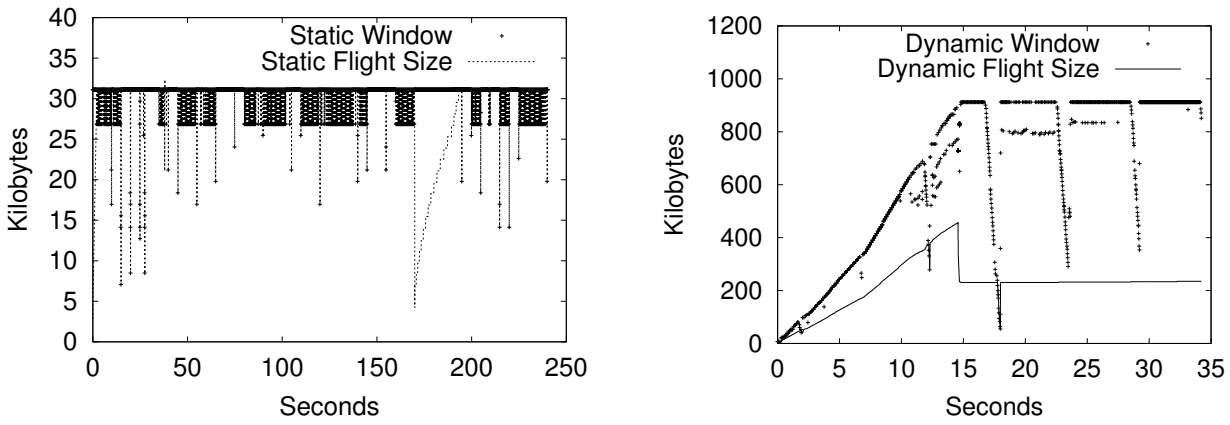
Because of their importance to the high-energy nuclear physics and other communities, we propose implementing user-space DRS in `bbcp` and `bbftp` to eliminate a major need for active probing of the network, to make the network capacity currently being consumed by active probing available for data transfers, and to relieve scientists of the burden of manually determining the appropriate buffer sizes for good performance.

We have chosen `bbcp` and `bbftp` over implementing DRS in GridFTP for several reasons. First, we are already working with the GridFTP developers to incorporate DRS. Modifying other applications brings the benefits of DRS to a larger scientific audience. Second, `bbcp` and `bbftp` are deployable by the scientists themselves without requiring administrative privileges. Third, `bbcp` and `bbftp` have most, if not all, the features of GridFTP, including restarting failed transfers, displaying progress messages, allowing buffer size specification, using multiple parallel streams, allowing 3rd party copies, and are Grid enabled (i.e. can use Grid certificates). Third, `bbcp` (and `bbftp` to a large extent) have additional features (compared to GridFTP) to meet the needs of the HENP community: (1) does not require installation of the Globus Toolkit, (2) does not require Globus credentials, (3) is peer-to-peer (no server needs to be running), (4) uses well-know ports for data transfers enabling easy detection, monitoring, and measurement of data flows, (5) uses MD5 checksums to verify correctness, (6) allows compression, (7) allows QoS specifications in the form of maximum transfer rates, (8) has the ability to deal with firewalls, (9) is familiar to `scp` users, and (10) has an object-oriented design and implementation that facilitates modification (e.g. to add DRS). Finally, the principle developer of `bbcp` is an investigator on the proposal and hence we can ensure that the DRS modifications will be adopted into the source tree.

2 Preliminary results

As a result of setting the buffer sizes to the estimated bandwidth-delay product of a connection, DRS enables dramatically higher throughput. Figure 2 shows the buffer sizes and unacknowledged data (flight) sizes of a test with and without kernel-space DRS. Figure 2(a) shows that statically limiting the buffer sizes to the operating system default, in this case 32 KB, allows only 32 KB of data to be in flight at a given time. In contrast, Figure 2(b) shows that DRS allows significantly more data to be in flight over the same network path.

At SC2001, we performed a live demonstration of DRS by transferring large files back and forth from the showroom floor to our facility in Los Alamos. When the showroom network was lightly loaded, e.g. at 7:30 AM before the doors opened, the FTP transfer rate with DRS was almost 30 times higher than the transfer rate of FTP over stock (untuned) TCP. During periods of heavy load, e.g. during the middle of the day, the transfer rate for FTP over both DRS and over stock TCP was about 4 Mbps. Not only does this demonstration show that automatic buffer adaptation via DRS delivers dramatically better performance, it also shows that DRS is "TCP-friendly." The demonstration also shows that the throughput increase



(a) Default Buffer Size

(b) Dynamic Right-Sizing

Figure 2: Window and Flight Sizes over Fast Ethernet

enabled by DRS depends upon network congestion. Typical improvements are in the range of 6–8 times stock TCP on networks with average load.

The performance improvements of DRS in user space is also significant. To demonstrate the performance improvements possible with drsFTP, we present results comparing the transfer of data across a emulated WAN with a RTT of 102.1 ms. We use the characteristics of the connection of Figure 1 to represent the WAN in the experiments. (The cumulative distribution function of the data in Figure 1 is shown in Figure 3. Several representative values are called out on the graph. In particular, the 0.5-Mb value corresponds to a buffer size of 64-KB, which is the default (stock) buffer size. Also, the 1.74-Mb value corresponds with one of the statically-set buffer sizes tested.)

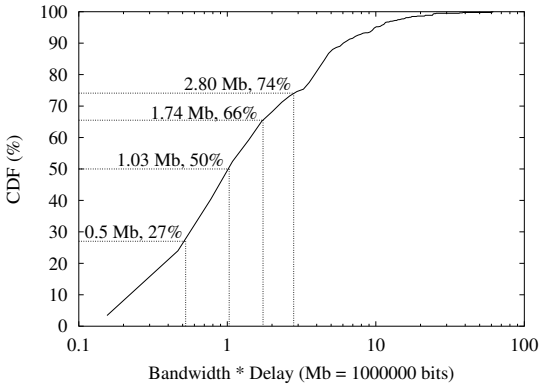


Figure 3: CDF of BDP from Figure 1

Figure 4 shows the average FTP bandwidth as a function of the size of the transfer. The average bandwidth of FTP with stock buffer sizes approaches 5 Mbps for file sizes as small as 8 MB. In contrast, the average bandwidth of drsFTP asymptotically approaches 30 Mbps at over 64 MB file transfers. Thus, the utilization of available bandwidth of drsFTP is approximately six times better than stock FTP.

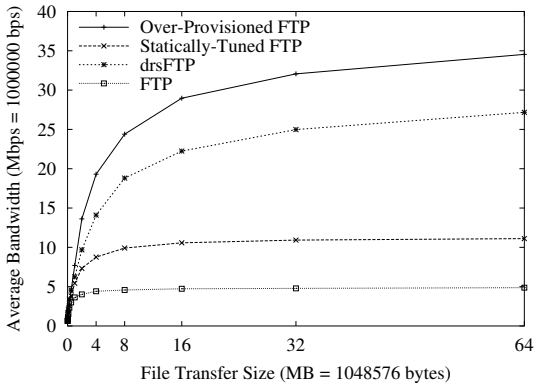


Figure 4: Comparison of FTP, drsFTP and statically-tuned FTP

The best bandwidth (34.5 Mbps) is achieved by the over-provisioned FTP which has larger than required buffer sizes. As shown, drsFTP achieves 78.7% of the over-provisioned bandwidth. The primary reason for the difference in performance is that drsFTP must rely on coarse-grained measurements to infer available bandwidth and round-trip time and hence may not infer the required buffer sizes accurately. This is an inherent limitation indicative of the interim nature of user-space DRS. The kernel-space version of DRS has access to fine-grained information and hence performs better in general [5, 6]. In addition, all applications benefit without modification when DRS is in the kernel.

Figure 4 also compares the average bandwidth of drsFTP to a statically-tuned case where the BDP was sampled at an inopportune time, e.g. at one of the lower data points in Figure 1. As shown in Figure 3, the 212.5 KB (1.74 Mb) buffer size chosen for this test is in the 66th percentile of the BDP data of Figure 1. Here we see that drsFTP utilizes the available bandwidth 2.4 times better than the statically-tuned case. The comparison illustrates the benefit of inferring the available bandwidth and setting the flow-control buffers automatically throughout the lifetime of a connection. By not relying upon the estimated bandwidth-delay product obtained before the data transfer commences, drsFTP is able to perform better.

3 Proposed Work

Through discussions with colleagues in the physical sciences, particularly colleagues in high-energy nuclear physics, we have come to the conclusion that the most direct way to improve network performance for scientists is to make DRS available in as many contexts as possible. Towards this end, we have been talking with open source and proprietary operating system vendors in order to incorporate DRS into their OS kernels. Simultaneously, we are making DRS more widely available by increasing the number of applications that support DRS. Although the former is "the right thing to do" in the long run, the later is the primary focus of this proposal.

We propose implementing user-space DRS in `bbcp` and `bbftp`. We have chosen these applications because of their importance to colleagues in the high-energy nuclear physics community. Implementing DRS in user space will largely eliminate active probing of the network for the purpose of tuning TCP bulk data transfers. This will make more network capacity available for data transfers, as well as relieve the

burden of manually determining the appropriate buffer sizes for good performance.

Unlike the kernel-space implementation of DRS which benefits all applications transparently, each application must have its own user-space implementation of DRS. There are two approaches: (1) implement DRS separately for `bbcp` and `bbftp`, or (2) develop a library implementation of DRS that reduces the modifications needed to implement DRS in applications. In order to benefit the high-energy nuclear physics community as quickly as possible, we will directly implement DRS in one of the applications in the first year. In the second year, we will design and implement a DRS library (called `libdrs`) in order to facilitate adding DRS to any application, not just `bbcp` and `bbftp`. By the third year, we will release production-ready versions of `bbcp` and `bbftp` built upon `libdrs`.

Using a two pronged approach quickly puts the improved performance of DRS into the hands of scientists while reducing the effort of adding DRS to other applications in the long term. Reducing the cost of incorporating DRS greatly increases the number of applications that will be modified and increases the positive impact on the scientific community. It also allows us to better reach our goal of helping scientists forget about network issues.

Currently, bulk-data transport applications assume TCP as the underlying transport protocol. Future high-performance networks, such as UltraScienceNet [19] and UltraLight [20] will be based upon circuit-switching of optical paths. The circuit-switched nature of UltraScienceNet ensures that only a single pair of hosts can use a light path at one time. Consequentially, there is no need for the congestion-control and slow-start mechanisms of TCP. The routing functionality of IP is also redundant. Production operating systems, however, do not allow portions of TCP/IP to be disabled and therefore cannot take advantage of a dedicated light path to deliver increased performance to the application. The ability to disable functionality would require kernel modifications that are not practical for many systems. We propose adding support in `bbcp` and `bbftp` for such networks.

The proposal also includes evaluating and comparing the DRS solutions with alternate solutions such as new advanced TCP stacks, (e.g. FAST [11], HSTCP-LP [12], HS-TCP [13], Scalable TCP [14]), reliable UDP based protocols such as UDT [15], as well as some special file tranfer applications such as SOBAS [16] or RBUDP [17]. We will evaluate and compare the fairness, stability, throughput performance, and cpu utilization, as well as the ease of use, ease of configuration, capabilities, flexibility, integration, and robustness. Feedback from the evaluations will be used to help guide improvements in the DRS (and other transport mechanisms) and file copy/transfer solutions. The evaluation will be performed using the IEPM infrastructure [18] that provides ssh access to use high speed (100 Mbps to 1 Gbps) production paths between over 40 hosts in 9 countries. In addition we will also use high performance (10 Gbps) testbeds such as the DoE UltraScienceNet [19], and UltraLight [20].

Finally, we will investigate incorporating credential-based authentication and authorization mechanisms in `bbcp` and `bbftp` in order to reduce the administrative burden on scaling to large numbers of users.

4 Research Issues

Although DRS delivers dramatically better throughput by ensuring that buffer sizes are not the limiting factor, there are several issues that need to be explored. We plan to systematically address these issues as

appropriate.

First, we have ignored the effects of buffer management on the performance of DRS and on the fairness of buffer utilization on end hosts. It was shown in [8] that dramatic improvements in performance can be obtained by wisely designing buffer management strategies. In addition, we need to quantify the interaction between DRS and Linux buffer management policies. It is critical for us to thoroughly demonstrate that there are no harmful interactions between DRS and existing Linux buffer management techniques, regardless of whether or not DRS is in kernel or user space. Because buffers cannot be directly managed in user space, influencing buffer management in user space will be a challenge.

Another issue concerns excessive buffer usage. Currently, DRS sets the buffer sizes to be twice the estimated size of the sender's congestion-control window in order to ensure that the connection is not throttled during slow start. By taking note of when TCP exits slow start, we should be able to cut the buffer usage of DRS in half in equilibrium. Since congestion control is performed on the sender, the receiver has no way of knowing when slow start ends. We will therefore need to develop a technique for determining this event.

There is another way in which buffer usage can be reduced. Figure 1 shows that the bandwidth-delay product, and hence amount of buffer space required on the receiver, varies dramatically with time. Näively reducing buffer space can cause massive packet loss (and hence poor performance) because the sender may already have packets in flight that will have to be dropped if buffer space is not available. To avoid this, DRS monotonically increases buffer sizes. The conditions under which buffer usage on the receiver can be reduced needs to be determined. The benefit is more memory available to support individual applications or to support more concurrent applications.

Another issue to be explored in user-space DRS implementations is the effect of scheduling on buffer sizes and on DRS policies. We have observed user-space DRS out performed kernel-space DRS. We hypothesize that user-space DRS naturally accounts for the additional delay caused by OS scheduling in the way it estimates RTT. This fortunate anomaly needs to be thoroughly investigated and kernel-space DRS needs to be extended to account for scheduling effects, if valid.

Finally, although theoretical arguments and anecdotal evidence indicate that DRS is TCP-friendly, we have yet to show experimental evidence that such is the case. We will do so as part of the project.

5 Timeline and Deliverables

The timeline for the proposed work is

- Year 1
  - Analyze the architectures of `bbcp` and `bbftp` to become familiar with their designs. (This will also extend our understanding of how DRS should be implemented as a universal user-space library.)
  - Choose one of the applications and directly implement DRS.
  - Test in the laboratory on WAN emulators, on production networks such as ESnet and Abilene, and on high performance testbeds such as the DoE UltraScienceGrid and UltraLight, to ensure

correctness and performance goals are met.

  - Make the interim implementation available for use by scientists.

- Year 2

  - Design and implement the universal DRS library, `libdrs`.
  - Modify `bbcp` and `bbftp` to use `libdrs`, modifying `libdrs` as needed to be more universal.
  - Test in the laboratory on WAN emulators and on production networks such as ESnet and Abilene, and on high performance testbeds such as the DoE UltraScienceGrid and UltraLight, to ensure correctness and performance goals are met.
  - Enlist the aid of colleagues to beta test software in scientific environments.
  - Modify applications and library as needed based upon feedback.
  - Investigate approaches to adding certificate-based authentication and authorization to `bbcp` and `bbftp`. Begin implementation.

- Year 3

  - Release modified `bbcp` and `bbftp` applications.
  - Release production-ready `libdrs` implementation.
  - Feed modifications to `bbcp` and `bbftp` back into the main source trees.
  - Set up infrastructure (possibly SourceForge) for supporting `libdrs`.
  - If time permits, look at improving the DRS algorithm for both kernel- and user-space implementations.
  - If time permits, look at developing a library for authentication / authorization to extend `bbcp` and `bbftp` to support kerberos and PKI in much the same way that `libdrs` makes it easier to support DRS.

The software artifacts resulting from the project are (1) a universal library implementation of DRS (`libdrs`) released under an open source license, (2) a modified implementation of `bbcp` supporting user-space DRS, and (3) a modified implementation of `bbftp` also supporting user-space DRS. In addition, we expect to publish several publications on the adaptation of DRS to a user-space library, the performance improvements achieved by using `libdrs`, and any improvements to the DRS algorithm as the result of the research effort. Finally, if the adapted software begins improving the delivered performance in daily scientific discourse, as we expect, we will consider the project a success.

6 Deployment

Our initial deployment will be within BaBar, as we have direct ties to physicists with the project. However, we also have ties with other DOE SciDAC funded projects that will enable us to deploy the software more widely for the benefit of the scientific community. The first is the Terascale Supernova

Initiative (TSI) [21] due to LANL's ties with the PI of the project. The second is the Supernova Science Center (SSC) [22] due to LANL's previous work with the Dr. Michael S. Warren in creating the energy-efficient Green Destiny cluster, a "2003 R&D 100 Award" winner. **FIX ME: Any others?**

7 Conclusions

The majority of the traffic on the DOE Abilene network is related to transfering data between sites. In order to achieve good performance, TCP send and receive buffers need to be tuned. Traditionally, the buffer sizes have been tuned by hand. This is tedious and wasteful of scientists' time. The dynamic right-sizing (DRS) algorithm automatically and continuously modifies the buffer sizes according to network conditions. As a result, it achieves dramatic improvements in performance without requiring attention by the scientists.

Based upon the needs of the high-energy physics community, we propose to implement DRS in two popular data transfer applications, `bbcp` and `bbftp`. We plan to take a two phase approach. The first is to implement DRS directly in one of the two programs by the end of the first year to make DRS available to scientists as quickly as possible. The second is to develop a library-based implementation of DRS, called `libdrs`, to make it easier to implement DRS in all user-space applications. By the end of three years, we will have developed and deployed `bbftp` and `bbcp` in the high-energy nuclear physics community. Along the way, we intend to solve open issues with DRS to improve its performance further. We will also add support for operating over circuit-switched optical networks, such as UltraSciencesNet and UltraLight. Finally, we will investigate adding support for certificate-based authentication and authorization to improve security.

8 Biographical Sketches

8.1 Wu-chun Feng

Computer and Computational Sciences Research Division	Tel: 505-655-2730
Los Alamos National Laboratory	Fax: 505-665-4934
P.O. Box 1663, M.S. D451	Email: feng@lanl.gov
Los Alamos, NM 87545	

Dr. Wu-chun Feng is a technical staff member and team leader of Research & Development in Advanced Network Technology (RADIANT) in the Advanced Computing Laboratory (CCS-1) at Los Alamos National Laboratory. Dr. Feng joined LANL in 1998 and has held adjunct assistant professorships at The Ohio State Univ. (2000-2003) in the Dept. of Computer & Information Sciences and Purdue Univ. in the Dept. of Electrical & Computer Engineering (1998-2000). Prior to LANL, he was an assistant professor at the Univ. of Illinois at Urbana-Champaign in the Dept. of Computer Science. While at the Univ. of Illinois, he was also involved in a successful start-up company called Vosaic, which produced software for real-time streaming of audio and video over the Internet. He has also had professional stints at NASA Ames Research and IBM T.J. Watson Research Center.

Dr. Feng brings a wide breadth and depth of knowledge to the table, as indicated by his research portfolio of over 70 peer-reviewed publications, most in the last three years. His current research interests span high-performance networking and computing, network monitoring and measurement, active queue management in routers, cyber-security, and bioinformatics. Highlights of his recent professional activities and appointments include [1] Program Chair of the 2005 Int'l Conf. on Parallel Processing, [2] Program Vice-Chair at the 1999 Int'l Conf. on Parallel Processing, [3] Program Committee Member for IEEE/SC 2000, IEEE Int'l Symp. on High-Perf. Distrib. Computing (2001, 2003), IEEE Int'l Parallel & Distrib. Proc. Symp. (2003), and IEEE Conf. on Local Computer Networks (2001-2003), [4] IEEE Distinguished Speaker (IEEE Distinguished Visitor's Program. See `http://www.computer.org/chapter/DVP/Northamerica.htm`.), and [5] fifteen invited talks and colloquia in the last two years.

Education

- Ph.D. in Computer Science, University of Illinois at Urbana-Champaign, 1996.
- M.S. in Computer Engineering, Penn State University, 1990.
- B.S. in Electrical & Computer Engineering and Music, Penn State University, 1988.

Selected Publications

- M. Gardner, S. Thulasidasan, and W. Feng, "User-Space Auto-Tuning for TCP Flow Control in Computational Grids," *Computer Communications*, 2004.
- A. Engelhart, M. Gardner, and W. Feng, "Re-Architecting Flow-Control Adaptation for Grid Environments," *18th IEEE International Parallel & Distributed Processing Symposium*, April 2004.

- S. Ayyorgun and W. Feng, "A Deterministic Characterization of Network Traffic for Average Performance Guarantees," *38th Annual Conference on Information Scinces and Systems*, March 2004.

- J. Hurwitz and W. Feng, "End-to-End Performance of 10-Gigabit Ethernet on Commodity Systems," *IEEE Micro*, January-February 2004.

- W. Feng, M. Gardner, M. Fisk, and E. Weigle, "Automatic Flow-Control Adaptation for Enhancing Network Performance in Computational Grids," Journal of Grid Computing, Vol. 1, No. 1, 2003.

- W. Feng, J. Hurwitz, H. Newman, S. Ravot, L. Cottrell, O. Martin, F. Coccetti, C. Jin, D. Wei, and S. Low, "Optimizing 10-Gigabit Ethernet for Networks of Workstations, Clusters, and Grids: A Case Study," SC 2003: High-Performance Networking and Computing Conference, Phoenix, AZ, November 2003.

- M. Veeraraghavan, X. Zheng, H. Lee, M. Gardner, and W. Feng, "CHEETAH: Cicuit-Switched High-Speed End-to-End Transport ArcHitecture," Best Paper Award, *SPI/IEEE Optical Networking and Computing Communications Conference*, October 2003.

- W. Feng, "Making a Case for Efficient Supercomputing," *ACM Queue*, October 2003.

- J. Hurwitz and W. Feng, "Initial End-to-End Performance Evaluation of 10-Gigabit Ethernet," *IEEE Hot Interconnects*, August 2003.

- S. Thulasidasan, W. Feng, and M. Gardner, "Optimizing GridFTP Through Dynamic Right-Sizing," *IEEE Symposium on High-Performance Distributed Computing*, June 2003.

- W. Feng and S. Vanichpun, "Ensuring Compatibility Between TCP Reno and TCP Vegas," *IEEE Symposium on Applications and the Internet*, Orlando, FL, January 2003.

- W. Feng, M. Fisk, M. Gardner, and E. Weigle, "Dynamic Right-Sizing: An Automated, Lightweight, and Scalable Technique for Enhancing Grid Performance," *Lecture Notes in Computer Science*, Vol. 2334, 2002. (A preliminary version of this paper appeared in the 7th IFIP/IEEE Workshop on Protocols for High-Speed Networks.)

- S. Vanichpun and W. Feng, "On the Transient Behavior of TCP Vegas," IEEE International Conference on Computer Communications and Networks (IC3N'02), October 2002.

- M. Gardner, W. Feng, and M. Fisk, "Dynamic Right-Sizing in FTP (drsFTP): An Automatic Technique for Enhancing Grid Performance," *IEEE Symposium on High-Performance Distributed Computing*, July 2002.

- E. Weigle and W. Feng, "A Comparison of TCP Automatic-Tuning Techniques for Distributed Computing," *IEEE Symposium on High-Performance Distributed Computing*, July 2002.

- F. Petrini, W. Feng, A. Hoisie, S. Coll, and E. Frachtenberg, "The Quadrics Network (QsNet): High-Performance Clustering Technology", *IEEE Micro*, January-February 2002.

- E. Weigle and W. Feng, "Dynamic Right-Sizing in TCP: A Simulation Study," *IEEE International Conference on Computer Communications and Networks*, October 2001.

- E. Weigle, W. Feng, and M. Gardner, "Why (Current) TCP Will Not Scale to the Next-Generation Internet," 11th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2001), Boulder, CO, March 2001.

- W. Feng, "The Future of High-Performance Networking," Workshop on New Visions for Large-Scale Networks: Research and Applications (Sponsors: Federal Large Scale Networking Working Group, DARPA, DOE, NASA, NIST, NLM and NSF), March 2001.

- W. Feng and P. Tinnakornsrisuphap, "The Failure of TCP in Distributed Computational Grids," SC 2000: High-Performance Networking and Computing Conference, Dallas, TX, November 2000.

- W. Feng and P. Tinnakornsrisuphap, "The Adverse Impact of the TCP Congestion-Control Mechanism in Heterogenous Computing Systems," International Conference on Parallel Processing (ICPP 2000), Toronto, Canada, August 2000.

Invited Talks and Colloquia

- Bridging the Disconnect Between the Network and Large-Scale Scientific Applications, ACM SIGCOMM Workshop on Network-I/O Convergence: Experiences, Lessons, and Implications (NICELI), August 2003.

- Systems & Network Research for Grids, Argonne National Laboratory, October 2002.

- The Future of High-Performance Networking, Rice University, January 2002.

- The Future of High-Performance Networking, Workshop on New Visions for Large-Scale Networks: Research and Applications (Sponsors: Federal Large Scale Networking Working Group, DARPA, DOE, NASA, NIST, NLM and NSF), March 2001.

- The Failure of TCP over High-Performance Computational Grids, University of Illinois at Urbana-Champaign, January 2001.

- Network Traffic Characterization of TCP in Distributed Computational Grids, University of Oregon, June 2000.

Recent Awards and Honors

- Asian-American Engineer of the Year, February 2004.

- Sustained Bandwidth Award (a.k.a. "Moore's Law Move Over!" Award), SC2003 Bandwidth Challenge, November 2003.

- R&D 100 Award for Green Destiny, October 2003.

- Best Paper Award, IEEE Optical Networking and Computer Communications Conference (OptiComm), October 2003.
- Distinguished Performance Award, September 2003.
- Distinguished Mentor Performance Award, August 2003.

8.2 Mark K. Gardner

Computer and Computational Sciences Research Division
Advanced Computing Laboratory, CCS-1
Los Alamos National Laboratory
P.O. Box 1663, M.S. D451
Los Alamos, NM 87545

Tel: 505-655-4953
Fax: 505-665-4934
Email: mkg@lanl.gov
WWW: `http://public.lanl.gov/mkg/`

Dr. Mark K. Gardner is a technical staff member in the Research and Development in Advanced Network Technology (RADIANT) team of the Computer and Computational Sciences Division at Los Alamos National Laboratory. His research has focused on OS-bypass protocols, automatic flow-control adaptation for TCP, monitoring and measuring tools for the end hosts of high-speed wide-area networks, network traffic characterization, stochastic analysis of computer systems and program compilation.

Research Interests

- High-performance networking, network protocols, network traffic characterization, operating system design and program compilation.

Education

- Ph.D in Computer Science, University of Illinois at Urbana-Champaign, 1999.
- M.S in Computer Science, Brigham Young University, 1994.
- B.S in Mechanical Engineering, Brigham Young University, 1986.

Current Position

- Technical Staff Member, Research and Development in Advanced Network Technology (RADIANT), Los Alamos National Laboratory, 1999-Present.

Professional Experience

- Applications Researcher, U.S. Army Construction Engineering Research Laboratory, 1995. Automated decision support systems.
- Aerodynamicist, Allied-Signal Aerospace, Garrett Auxiliary Power Division, 1986-1991. Jet engine research and development.

Selected Professional Activities

- DOE SBIR proposal reviewer.
- NSF proposal reviewer.
- Reviewer for Real-Time Systems Journal; Software Practice and Experience.

Selected Publications

- A. Engelhart, M. Gardner, and W. Feng, "Re-Architecting Flow-Control Adaptation for Grid Environments," 18th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2004), Santa Fe, New Mexico, 26–30 April 2004 (to appear).
- M. Gardner, W. Deng, C. Mendes, W. Feng and D. Reed, "A High-Fidelity Software Oscilloscope for Globus," GlobusWORLD 2004, San Francisco, California, January 2004.
- M. Gardner, S. Thulasidasan, and W. Feng, "User-Space Auto-Tuning for TCP Flow Control in Computational Grids," Computer Communications, Elsevier, accepted for publication, 2003.
- W. Feng, M. Gardner, M. Fisk, and E. Weigle, "Automatic Flow-Control Adaptation for Enhancing Network Performance in Computational Grids," Journal of Grid Computing, Vol. 1, No. 1, 63–74, 2003.
- M. Veeraraghavan, X. Zheng, H. Lee, M. Gardner, and W. Feng, "CHEETAH: Circuit-Switched High-Speed End-to-End Transport ArcHitecture," 4th SPIE/IEEE Optical Networking and Computer Communications Conference (OptiComm 2003). Best paper award. Dallas Texas, 15–18 October 2003
- S. Thulasidasan, M. Gardner, and W. Feng, "Optimizing GridFTP through Dynamic Right-Sizing," 12th IEEE Symposium on High-Performance Distributed Computing (HPDC-12/2003), Seattle Washington, June 2003.
- M. Gardner, W. Feng, M. Broxton, A Engelhart, and G. Hurwitz, "MAGNET: A Tool for Debugging, Analysis and Adaptation in Computing Systems," 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid'2003), Tokyo, Japan, May 2003.
- M. Gardner, M. Broxton, A. Engelhart, and W. Feng, "MUSE: A Software Oscilloscope for Clusters and Grids," 17th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2003), Nice, France, April 2003.
- W. Feng, M. Gardner, and J. Hay, "The MAGNeT Toolkit: Design, Evaluation, and Implementation," Journal of Supercomputing, Vol. 23, No. 1, August 2002, pp. 67-79.
- M. Gardner, W. Feng, and M. Fisk, "Dynamic Right-Sizing in FTP (drsFTP): An Automatic Technique for Enhancing Grid Performance," 11th IEEE Symposium on High-Performance Distributed Computing (HPDC-11/2002), Edinburgh, Scotland, July 2002.

- W. Feng, M. Fisk, M. Gardner, and E. Weigle, "Dynamic Right-Sizing: An Automated, Lightweight, and Scalable Technique for Enhancing Grid Performance," 7th International IEEE Workshop on Protocols for High Speed Networks (PfHSN 2002), Berlin, Germany, April 2002.
- M. Gardner, W. Feng, and J. Hay, "Monitoring Protocol Traffic with a MAGNeT," Passive and Active Measurement Workshop (PAM2002), Fort Collins, Colorado, March 2002.
- J. Hay, W. Feng, and M. Gardner, "Capturing Network Traffic with a MAGNeT," 5th Annual Linux Showcase and Conference (ALS'01), Oakland, California, November 2001.
- W. Feng, J. Hay, and M. Gardner, "MAGNeT: Monitor for Application-Generated Network Traffic," 10th International Conference on Computer Communication and Networking (IC3N'01), Scottsdale, Arizona, October 2001.
- E. Weigle, W. Feng, and M. Gardner, "Why (Current) TCP Will Not Scale to the Next-Generation Internet," 11th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2001), Boulder, CO, March 2001.

Invited Talks and Colloquia

- Facing the Future of Ubiquitous Network Bandwidth, Brigham Young University, January 2001.
- Scheduled Transfer: An ANSI OS-Bypass Protocol, University of Utah, February 2001.

8.3 Roger Leslie Anderton Cottrell

Stanford Linear Accelerator Center	Tel: 650-926-2523
2575 Sand Hill Road, Mail Stop 97	Fax: 650-926-3329
Stanford, California 94309	Email: cottrell@stanford.edu

Dr. Cottrell joined SLAC as a research physicist in High Energy Physics, focusing on real-time data acquisition and analysis in the Nobel prize winning group that discovered the quark. In 1972–73, he spent a year's leave of absence as a visiting scientist at CERN in Geneva, Switzerland, and in 1979–80 at the IBM U.K. Laboratories at Hursley, England, where he obtained United States Patent 4,688,181 for a dynamic graphical cursor. He is currently the Assistant Director of the SLAC Computing Services group and lead for the computer networking and telecommunications areas. He is also a member of the Energy Sciences Network Site Coordinating Committee (ESCC) and the chairman of the ESnet Network Monitoring Task Force. He is also the leader/PI of the DoE sponsored Internet End-to-end Performance Monitoring (IEPM) effort, and the ICFA network monitoring working group. He was a leader of the effort that, in 1994, resulted in the first Internet connection to mainland China.

Employment Summary

- Assistant Director of Computing Services, Stanford Linear Accelerator Center; manage networking and computing; 1982 to present.
- Manager of SLAC Computer Network, Stanford Linear Accelerator Center; manage all SLAC's computing activities; 1980–82.
- Visiting Scientist, IBM U.K. Laboratories, Hursley, England; graphics and intelligent distributed workstations; 1979–80.
- Staff Physicist, Stanford Linear Accelerator Center; Inelastic e-p scattering experiments, physics and computing; 1967–79.
- Visiting Scientist, CERN; Split Field Magnet experiment; 1972–73.

Education Summary

- Manchester University, Ph.D., Interactions of Deuterons with Carbon Isotopes, 1962–67.
- Manchester University, B.Sc., Physics, 1959–62.

Publications

The full list of 70 publications is readily available from online databases. Included here is only a limited number of recent publications relevant to networking.

- H. Cerdeira, E. Canessa, C. Fonda, and R. L. Cottrell, "Developing Countries and the Global Science Web," CERN Courier December 2003.
- R. Les Cottrell and Warren Matthews, "Measuring the Digital Divide with Pinger," Developing Countries Access to Scientific Knowledge: Quantifying the Digital Divide, ICTP Trieste, October 2003; also SLAC-PUB-10186.
- R. Les Cottrell and Connie Logg, "Pinger History & Methodology," Developing Countries Access to Scientific Knowledge: Quantifying the Digital Divide, ICTP Trieste, October 2003; also SLAC-PUB-10187.
- R. Les Cottrell and Enrique Canessa, "Internet Performance to Africa," Developing Countries Access to Scientific Knowledge: Quantifying the Digital Divide, ICTP Trieste, October 2003; also SLAC-PUB-10188.
- Vinay Ribeiro, Rudolf Reidi, Richard Baraniuk, Jiri Navratil, and Les Cottrell, "PathChirp: Efficient Available Bandwidth Estimaion for Network Paths," Passive and Active Measurements (PAM) Workshop 2003, April 2003; also SLAC- PUB-9732.
- E. Canessa, H. A. Cerdeira, W. Matthews, and R. L. Cottrell, "Monitoring the Digital Divide," Computing in High Energy Physics and Nuclear Physics (CHEP 2003), San Diego, March 2003; also SLAC-PUB-9730.

- R. Les Cottrell, Antony Antony, Connie Logg and Jiri Navratil, IGRID2002 Demonstration Bandwidth from the Low Lands," Future Generation Computer Systems 19 (2003) 825–837, Elsevier Science B.V.; also SLAC-PUB-9560, October 31, 2002.

- Connie Logg and Les Cottrell, "Passive Performance Monitoring and Traffic Characteristics on the SLAC Internet Border," to appear in Computing in, High-Energy Physics and Nuclear Physics (CHEP 2001), Beijing, China, 3–7 Sep 2001; also SLAC-PUB-9174, 4pp.

- Warren Matthews, Les Cottrell, and Davide Salomoni, "Passive and Active Monitoring on a High Performance Research Network," Passive and Active Measurements (PAM) Workshop 2001, Amsterdam, April 22–24 2001; also SLAC-PUB-8776, Feb 2001. 6pp.

- W. Matthews and L. Cottrell, "The Pinger Project: Active Internet Performance Monitoring for the HENP Community," IEEE Commun. Mag. 38:130–136, 2000; also SLAC-REPRINT-2000-008, May 2000, 7pp.

- Warren Matthews, Les Cottrell, and Charles Granieri, "International Network Connectivity and Performance, the Challenge from High-Energy Physics," talk presented at the Internet2 Spring Meeting, Washington D.C., 27 Mar 2000; also SLAC-PUB-8382, Mar 2000, 18pp.

- Warren Matthews and Les Cottrell, "Internet End-to-End Performance Monitoring for the High-Energy Nuclear and Particle Physics Community," Passive and Active Measurement (PAM) Workshop Hamilton, New Zealand, 3–4 Mar 2000; also SLAC-PUB-8385, Feb 2000, 10pp..

- W. Matthews and L. Cottrell,"Pinger: Internet Performance Monitoring: How No Collisions Make Better Physics," International Conference on Computing in High Energy Physics and Nuclear Physics (CHEP 2000), Padova, Italy, 7-11 Feb 2000; also SLAC-PUB-8383, Feb 2000, 5pp..

- R. Les. Cottrell, "Discussant Remarks on Session: Statistical Aspects of Measuring the Internet," in Proceedings of the 30th Symposium on the Interface, (ISBN 1-886658-05-6).

- Warren Matthews, Les Cottrell, and David Martin, "Internet Monitoring in the HEP Community," International Conference on Computing in High-Energy Physics (CHEP 98; also SLAC-PUB-7961, Oct 1998. 8pp.

- R.L.A. Cottrell, C.A. Logg, D.E. Martin, "What is the Internet Doing for and to You?" talk given at Computing in High-energy Physics (CHEP 97), Berlin, Germany, 7-11 Apr 1997; also SLAC-PUB-7416, Jun 1997, 7pp.

Lecture Courses

- Les Cottrell, "How the Internet Works," International Nathiagali Summer College Lecture Course, Pakistan, Summer 2001.

9 Current and Pending Support

9.1 LANL

9.2 SLAC

- R. Les Cottrell

  * Current Support:

    Project: DOE/SciDAC Edge-based Traffic Processing and Service Inference for High-Performance Networks (INCITE)
    Percent support: 5%
    Duration: ends October 2004

    Project: DOE HENP base funding
    Percent support: 95%

  * Other Pending Support:

    Project: DOE/SciDAC TeraPaths: A QoS enabled Collaborative Data Sharing Infrastructure for Peta-scale Computing Research
    Percent support: 8%

    Project: DOE/SciDAC INCITE Ultra - New Protocols, Tools, Security, and Testbeds for Ultra High-Speed Networking
    Percent support: 8%

    Project: DoE/SciDAC Measurement and Analysis for the Global Grid and Internet End-to-end perfomance (MAGGIE)
    Percent support: 16%

    Project: European Commission/Sixth Framework Program: Gateway to Science and Development (GS&D): an innovative integrated approach to the enhancement of science in developing countries with applications to disaster prevention and evaluation of natural resources
    Percent support: 8%

    Project: NSF/NMI program: Middleware for Global Optimized Data Distribution in the Internet (GODDI)
    Percent support: 8%

10 Facilities and Resources

Between the two institutions, we already have the facilities and hardware resources needed to complete the project. We will use existing equipment to develop the DRS library, to add authentication services to

`bbcp`, and to modify the applications for DRS and authentication support. We will also use our existing WAN emulation infrastructure to test the performance and correctness of the software in the laboratory. For live WAN testing, we will use existing network connections from SLAC to Caltech, CERN, UvA, and BaBar sites throughout the world.

10.1 Los Alamos National Laboratory

The following compute resources are available to the researchers:

- Two 120-node Linux research clusters, with each node composed of a 1-GHz Transmeta processor with high-performance, code-morphing software, 640-MB memory and 20-GB hard disk, all connected by Fast Ethernet. (Each cluster only occupies six square feet.)
- A 240-node Linux research cluster, with each node composed of a 1-GHz Transmeta processor with high-performance, code-morphing software, 640-MB memory, 20-GB hard disk, and connected by Fast Ethernet. (This cluster only occupies six square feet.)
- An 11-node Linux research cluster, with each node composed of dual-processor AMD Opterons (eight nodes at 1.4 GHz; three nodes at 2.0 GHz), 1 GB memory, and connected by Gigabit Ethernet and 10-Gigabit Ethernet.
- An 18-node Linux research cluster with each node composed of a 1.13-GHz Intel Pentium III processor, 512-MB memory, 10-GB hard disk, and connected via Gigabit Ethernet.
- A 5-node Linux research cluster with each node composed of a 1-GHz VIA EPIA processor, 128-MB memory, and connected via Fast Ethernet. (This cluster only occupies a 0.5 ft. x 0.5 ft. x 1 ft. footprint.)
- An 9-node Linux research cluster with each node composed of a dual-processor 500-MHz Intel Pentium II, 1 GB memory, and connected via Gigabit Ethernet.
- A wide array of server-class compute nodes including three Dell PowerEdge 2650s (2.2-GHz dual P4 Xeons), three Dell PowerEdge 4600s (2.4-GHz dual P4 Xeons), and a pair of lower-power Itanium2 servers.

The following networking resources are available to the researchers:

- Six Intel 10-Gigabit Ethernet PCI-X adapters.
- One Extreme Networks Summit 7i Switch : 28-port Gigabit Ethernet (24 copper and 4 fiber).
- One 3Com SuperStack3 4924 Switch: 28-port Gigabit Ethernet (24 copper and 4 fiber).
- Two 3Com SuperStack3 Switch 4400: 48-port Fast Ethernet.
- One D-Link DGS-1024T Switch: 26-port Gigabit Ethernet (24 copper and 2 fiber).

- One D-Link DGS-1016T Switch: 18-port Gigabit Ethernet (16 copper and 2 fiber).

The following resources are also available:

- The Los Alamos Access Grid room with a projection wall that is 20 ft. wide by 10 ft. tall for distributed collaboration.
- An 18-projector (6 x 3) tiled display wall driven by a 32-node Linux cluster, of which only 18 nodes are used to drive the display (with the anticipation that the other nodes would allow us to perform parallel rendering/computing and Powerwall visualization).

10.2 Stanford Linear Accelerator Center

SLAC is the home of the BaBar HENP experiment. BaBar was recently recognized as having the largest database in the world. In addition to the large amounts of data, the SLAC site has farms of compute servers with over 3000 CPUs. The main (tier A) BaBar computer site is at SLAC. In addition, BaBar has major tier B computer/data centers in Lyon, France, near Oxford, England, Padova, Italy and Karlsruhe, Germany which share TBytes of data daily with SLAC. Further BaBar has 600 scientist and engineer collaborators at about 75 institutions in 10 countries. This is a very fertile ground for deployment and testing of bulk-data transfer utilizing improved TCP stacks and Grid replication middleware. There are close ties between the SLAC investigators, the BaBar scientists and the SLAC production network engineers.

The IEPM-BW infrastructure, developed at SLAC, has 10 monitoring sites in 5 countries and about 50 monitored sites with contacts, accounts, keys, software installed etc. This provides a valuable testbed for evaluating new TCP stacks etc. in a production environment. The SLAC IEPM group has a small farm of 6 high-performance network test hosts with 2.5- to 3.4-GHz CPUs and 10-GigE Network Interface Cards (NICs). SLAC hosts network measurement hosts from the following projects: AMP, NIMI, PingER, RIPE, SCNM, and Surveyor. SLAC has two GPS aerials and connections to provide accurate time synchronization.

SLAC has an OC-12 Internet connection to ESnet, and a 1-Gigabit Ethernet connection to Stanford University and thus to CalREN/Internet2. We have also set up experimental OC-192 connections to CalRENII and Level(3). The experimental connections are currently not in service, but have been successfully used at SC2000–2003 to demonstrate bulk-throughput rates from SuperComputing to SLAC and other sites at rates increasing over the years from 990 Mbits/s through 13 Gbps to 23.6 Gbps. SLAC is also part of the ESnet QoS pilot with a 3.5 Mbps ATM PVC to LBNL, and SLAC is connected to the IPv6 testbed with three hosts making measurements for the IPv6 community. SLAC has dark fibers to Stanford University and PAIX, SLAC plans to connect at 10 Gbps to the DoE UltraScienceNet and UltraLight testbeds later this year. The SLAC IEPM group has access to hosts with 10 Gbps connectivity at UvA on the NetherLight network in Amsterdam, at StarLight in Chicago and CERN in Geneva. We also have close relations with Steven Low's group at Caltech and plan to get access to their WAN-in-Lab setup for testing applications with dedicated long distance fiber loops. SLAC has been part of the SuperComputing bandwidth challenge for the last 3 years, part of the team that won the bandwidth challenge last year for the maximum data transferred and is a two time winner of the Internet2 Land Speed Record.

As part of our previous and continuing evaluations of TCP stacks, the SLAC IEPM team has close relations with many TCP stack developers (in particular the developers of FAST, H-TCP, HSTCP-LP, LTCP) and with the UDT developers.

11 References

[1] Internet2 NetFlow weekly reports: Week of 2004/03/08. `http://netflow.internet2.edu/weekly/20040308/`, March 8 2004.

[2] M. Allman, D. Glover, and L. Sanchez. Enhancing TCP over satellite channels using standard mechanisms. IETF RFC 2488, Jan 1999.

[3] B. Tierney. TCP tuning guide for distributed applications on wide-area networks. In *USENIX & SAGE Login*, February 2001. `http://www-didc.lbl.gov/tcp-wan.html`.

[4] J. Liu and J. Ferguson. Automatic TCP socket buffer tuning. In *Proceedings of SC 2000: High-Performance Networking and Computing Conference (Research Gem)*, November 2000. `http://dast.nlanr.net/Projects/Autobuf`.

[5] M. Fisk and W. Feng. Dynamic adjustment of tcp window sizes. Technical report, Los Alamos National Laboratory, 2000.

[6] M. Fisk and W. Feng. Dynamic Right-Sizing: TCP flow-control adaptation. In *Supercomputing 2001: High-Performance Networking and Computing Conference SC 2001*, November 2001.

[7] E. Weigle and W. Feng. Dynamic Right-Sizing: A simulation study. In *10th International Conference on Computer Communication and Networking (ICN 2001)*, Oct 2001.

[8] J. Semke, J. Mahdavi, and M. Mathis. Automatic TCP buffer tuning. *Computer Communications Review, ACM SIGCOMM*, 28(4), October 1998.

[9] M. Gardner, W. Feng, and M. Fisk. Dynamic Right-Sizing in FTP: Enhancing grid performance in user space. In *Proceedings of the IEEE Symposium on High-Performance Distributed Computing*, July 2002.

[10] S. Thulasidasan, W. Feng, and M. Gardner. Optimizing GridFTP through dynamic right-sizing. In *Proceedings of the IEEE Symposium on High-Performance Distribut ed Computing. (HPDC-12/2003)*, Jun 2003.

[11] C. Jin *et al.*. FAST Kernel: Background theory and experimental results. In *First International Workshop on Protocols for Fast Long-Distance Networks*, Feb 2003.

[12] A. Kuzmanovic, E. Knightly, and R. L. Cottrell. A protocol for low-priority bulk data transfer in high-speed high-rtt networks. In *PFLDnet 2004*, Feb 2004.

[13] S. Floyd. High speed TCP for large congestion windows. RFC 3649, Dec 2003.

[14] T. Kelly. Scalable TCP: Improving performance in highspeed wide area networks. In *First International Workshop on Protocols for Fast Long-Distance Networks*, Feb 2003.

[15] Y. Gu and R. Grossman. Udt: An application level transport protocol for grid computing. In *PFLDNet 2004*, 2004.

[16] R. Prasad, M. Jain, and C. Dovrolis. Socket buffer auto-sizing for high-performance data transfers. In *submitted to the Journal of Grid Computing*, 2004.

[17] E. He, J. Leigh, O. Yu, and T. DeFanti. Reliable blast udp: Predictable high performance bulk data transfer. In *IEEE International Conference on Cluster Computing (CLUSTER'02)*, Sep 2002.

[18] Internet end-to-end performance monitoring - bandwidth to the world (IEPM-BW) project. `http://www-iepm.slac.stanford.edu/bw/`.

[19] DOE UltraScience Net: Experimental ultra-scale network research testbed for large-scale science. `http://www.csm.ornl.gov/ultranet/`.

[20] Ultralight: An ultrascale information system for data intensive research. `http://ultralight.caltech.edu/`.

[21] Terascale Supernova Initiative. `http://www.phy.ornl.gov/tsi/`.

[22] Supernova Science Center. `http://www.supersci.org/`.

12 Budgets