DOE/MICS/Base Program Report for the National Collaboratory Middleware and Network Research Project Review

Project Title: High-Performance Transport Protocols

Project Type: Base

Principal Investigator: Wu-chun Feng (LANL)

Contact Information:
P.O. Box 1663
Research & Development in Advanced Network Technology (RADIANT)
Advanced Computing Laboratory (ACL) in the Computer & Computational Sciences Division
Los Alamos National Laboratory
Los Alamos, NM 87545
Tel: 505-665-2730
Fax: 505-665-4934
E-mail: feng@lanl.gov

This document presents a two-year progress report on a project entitled "High-Performance Transport Protocols" (HPTP). The ultimate goal of HPTP is to *automatically* deliver high-speed communication to application users over the wide-area network, thus providing a critical software piece of the network infrastructure that will better enable geographically separated scientists to effectively work together as a team and that will facilitate remote access to both facilities and data. At the present time, our techniques have been (or will be) incorporated into the following networking and middleware projects: GridFTP (DOE), LoCI (DOE SciDAC), Net100 (DOE), Web100 (NSF), FAST (NSF/AFOSR/ARO), and the FreeBSD operating system, the operating system used by Apple®. In addition, several large-scale scientific applications in distributed computing have already used (or have plans to use) our networking techniques, e.g., Supernova Science Center (DOE SciDAC), optIPuter (NSF ITR), LHC Computing Grid Project (CERN), and the U.S. Air Force.

1. Executive Summary

The explosive growth of long-distance, high-speed computer networks, combined with rapid and unpredictable developments in applications and workloads, has effectively crippled the "out-of-box" performance of today's ubiquitous transport protocol, TCP, in computational grids and distributed computing systems. Without tedious manual tuning of buffer sizes by network experts, TCP performance over the wide-area network (WAN) in support of distributed computational grids is so abysmal that visualization scientists can dump 150-GB of data to disk and then send it via Federal Express *three times faster* than electronically transferring the data via TCP over the 622-Mb/s Energy Sciences wide-area network between Los Alamos National Laboratory (LANL) and Sandia National Laboratory (SNL).

Even with manual tuning, performance can still be abysmal while wasting a tremendous amount of memory resources. To address this problem from a flow-control perspective, we propose a technique called *dynamic right-sizing* (DRS) that automates the sizing of memory buffers transparently *without* having to send out any extra probing traffic and without wasting memory resources. Our initial prototype, a patch to the Linux kernel, demonstrates a 15-fold improvement in throughput over a WAN while still abiding by TCP-friendliness and semantics. This DRS patch can be downloaded at http://public.lanl.gov/radiant/software/drs.html. (When used in parallel streams, DRS demonstrates over a 30-fold improvement in throughput.)

However, due to application scientists' unease with modifying an operating-system kernel, we received a number of requests to adapt DRS for utilization in user space, specifically in a bulk-data transfer utility like `ftp`. As a result, we researched and developed a lightweight drsFTP (i.e., dynamic right-sizing in FTP) alpha prototype in user space and also incorporated an alpha prototype of DRS into GridFTP [Allc01]. While these implementations currently deliver only a 6- to 8-fold improvement in throughput over a WAN, these gains in performance are achieved without any modification to the operating-system kernel.

Although the above research automatically improves network performance for large-scale scientific applications (as a standalone program or via GridFTP [Allc01], LoCI [Beck02], Net100 [Duni02], and Web100 [Web100]), it does so only from a flow-control perspective. Additional gains could be had by re-architecting TCP congestion control for large-scale scientific applications while simultaneously ensuring TCP fairness to other network flows, e.g., web traffic. Consequently, we are also collaborating with the FAST project [Jin03] at the California Institute of Technology to combine our DRS flow-control technique with their FAST congestion control, a relationship that materialized during our smashing of the Internet2 Land Speed Record in February 2003 (http://lsr.internet2.edu) [Feng03].

Finally, in addition to addressing fundamental transport issues directly, this project also strives to better *enable* transport protocols at the network and device-driver layers (via our research on 10-Gigabit Ethernet [Feng03, Hurw03]) and to better *support* middleware for grid services (via our MAGNET+MUSE research [Gard03a, Gard03b] that is part of a larger DOE SciDAC-funded project entitled "INCITE: Edge-based Traffic Processing and Service Inference for High-Performance Networks") and middleware for data transfer and management (via MAGNET+MUSE / drsFTP / GridFTP).

2. Goals & Technical Challenges

Over the past decade as networks with "large" bandwidth-delay products became more prevalent, large-scale scientific applications have needed a cadre of network wizards on-hand to troubleshoot and tune the network in support of their applications. The need for such wizards arguably started with the birth of cluster computing (also known as Beowulf clusters) back in 1994. Back then, troubleshooting and tuning the network, while still tedious, was at least a more tractable problem given that physical network resources were typically confined to a system-area network in a machine room. However, high-performance computing has evolved dramatically since then, and now we are faced with the significantly more difficult problem of troubleshooting and tuning the network over geographically disparate sites in support of distributed computing (e.g., computational grids, data grids, mobile- and sensor-based computing, peer-to-peer computing). Such a problem requires significant R&D investment to develop, integrate, and deploy a wide range of software tools and network capabilities. This is exactly the kind of investment that we are fortunate to have as part of the DOE Office of Science "National Collaboratory and High-Performance Network" research program.

At a high level, the goal of this project is simple: *automatically* provide reliable high-speed networking from *end-to-end*. To deliver high-speed networking automatically (i.e., remove the need for network wizards) to applications, we must thoroughly understand the troubleshooting and tuning processes that network wizards employ to achieve high speed and encapsulate those processes in the software/hardware cyberinfrastructure. For instance, although our breaking of the Internet2 Land Speed Record with TCP/IP in February 2003 took the effort of approximately 15 network wizards distributed across four disparate sites (i.e., Caltech, CERN, LANL, CERN), the exercise provided validation for our current line of research (e.g., dynamic right-sizing and "autotuning" 10-Gigabit Ethernet) as well as insight into future directions for automated network software (e.g., end-host monitoring tools that provide real-time status of distributed resources, an example of which can be found in the "Globus/ SvPablo/Autopilot/MAGNET+MUSE" suite of integrated software tools, the latter of which is part of the DOE SciDAC-funded INCITE project).

When we use the term "end-to-end" as part of the above stated goal, we do *not* mean it in the "SC Bandwidth Challenge" sense, i.e., to/from the network operation center (NOC) on the SC showroom floor. What we mean is from an application running on one end host to another application running on another end host across a local-area network, storage-area network, system-area network, or wide-area network in support of networks of workstations (e.g., Condor), network-attached storage, clusters, and grids, respectively. Given that these computing environments have fundamentally different characteristics, e.g., clusters run over dedicated rather than shared networks; TCP, as it exists today, is not the one-size-fits-all solution, but it could arguably evolve into such a solution.

In summary, the overarching technical challenge of this research is to provide the beginnings of a network infrastructure that will *automatically* and *transparently* provide high-speed network performance to end-host applications whether in a network of workstations, cluster, or grid. When appropriate, the network software will automatically adapt to its environment, e.g., disable congestion control when running in a dedicated environment such as a cluster. And when high speed is simply *not* possible, e.g., due to transient network congestion, the software infrastructure will provide the appropriate information (e.g., via MAGNET+MUSE) so that distributed scientific applications can adapt to the available resources whether those resources be compute, network, or storage resources.

3. Approach

Frustrated at the pitiful networking performance provided by the TCP/IP protocol suite in system-area networks (SANs) for clusters and supercomputers, researchers in high-performance computing proposed the notion of "user-level network interfaces" or ULNIs (also known as OS-bypass protocols) in the mid-1990s. These ULNIs reduced end-to-end latency by two orders of magnitude and increased realizable bandwidth by up to an order of magnitude in the SAN. The wild success of these high-performance ULNIs in SANs provided the impetus for new research directions including, but not limited to, the application of ULNI techniques to the I/O file system and proposals for a standard ULNI, e.g., Virtual Interface Architecture (VIA). However, to realize such performance at the application level, scientific applications needed to have an army of network wizards *and* network researchers on-hand to install, debug, and tune ULNI software over the system-area network. Thus, this high-performance solution was far out of the reach of many scientific applications.

Furthermore, the recent rise of grids for high-performance computing necessitates high-performance networking over the WAN, not just the SAN. While the performance of TCP/IP may be abysmal in the WAN, the protocol suite *seamlessly* scales from the SAN to the WAN. In contrast, while the performance of a ULNI is significantly better than TCP/IP, ULNI currently does not scale beyond the SAN, and furthermore, does not support congestion control. Thus, the high-performance networking community finds itself at a crossroad: Do we leverage our lessons learned in ULNI and incorporate them into TCP/IP (or a new high-performance transport protocol that sits on top of IP) in order to achieve better performance? Or do we "re-implement" portions of IP and TCP – specifically, routing and congestion control – into a ULNI so that it will scale to the WAN? Based on our extensive research experience with ULNIs (e.g., [Petr01]), research on multifractal network traffic analysis and modeling [Ribe00], congestion-control research such as [Jin03], and our SciDAC-funded research with Rice University and the Stanford Linear Accelerator Center on network inference and tomography, we believe that the answer is the former. In this project, we address the issue of transparently and

automatically providing high-speed networking to applications with a three-pronged attack: (1) dynamic right-sizing, (2) rude TCP, and (3) 10-Gigabit Ethernet.

Dynamic Right-Sizing: Currently, all commodity operating systems have their flow-control windows set statically by default to 32 KB or 64 KB. However, with bandwidth-delay products over the WAN ranging as high as 50 MB for a given network connection, this default setting of the flow-control window means that the network utilization over the connection can never exceed 0.1% despite the fact that there may be no congestion (or more specifically, little to no traffic) in the network. As a result, network wizards know to manually reset the flow-control window to 50 MB, for example, thus removing the artificial ceiling imposed by a 32- or 64-KB flow-control window and allowing the congestion-control window to regulate the connection as it should. However, when a connection is created in a local- or storage-area network where the bandwidth-delay product is on the order of tens of kilobytes, having the flow-control window set to 50 MB results in a potentially tremendous amount of memory being wasted, e.g., 10 KB / 50 MB = 0.01% memory utilization. Consequently, we propose a "dynamic right-sizing" of the flow-control window. By implementing such a mechanism within the specification of TCP, a "dynamically right-sized" TCP kernel will still be compatible with a stock TCP kernel. So, rather than continuing to assume that the receiver's advertised window (i.e., flow-control window) must be a static value, we allow the value to be dynamic and implement a receiver-side algorithm that infers what the sender's congestion window is and then advertises a flow-control window to the sender that is "right-sized" so that throughput, network utilization, and memory utilization are simultaneously optimized. Our initial results over a live WAN show that we can achieve upwards of a 15-fold improvement in throughput when the network is relatively uncongested; when the WAN becomes heavily congested, the congestion window will reflect the conditions appropriately and a "dynamically right-sized" connection will drop back and only take its fair share of the network bandwidth.

Rude TCP: When operating in an environment such as the DOE Science Grid or NSF TeraGrid, the networking environment from endpoint to endpoint may be dedicated or shared at any given time. For instance, in a cluster, say within the NSF TeraGrid, the network is a dedicated resource, and therefore, congestion control is irrelevant. In an enterprise grid, the network could be either dedicated or shared, thus congestion control should be applied only when the network is shared. Therefore, we propose *Rude TCP*, a variant of the ubiquitous TCP Reno protocol where congestion control can be automatically disabled and enabled, as appropriate, based on network conditions.

10-Gigabit Ethernet: Due in large part to our results with the Quadrics network [Petr02], we were asked by Intel to optimize the performance of their 10-Gigabit Ethernet cards. Optimizing such network interface cards is oftentimes a daunting task due to the low-level nature of the software. However, using the capabilities of MAGNET from our DOE SciDAC INCITE project, we were able to troubleshoot, tune, and optimize the performance of these cards in less than a week's time. In that week of time, we improved the "out-of-box" performance of the cards from about 2 Gb/s to over 4 Gb/s across a pair of low-end Dell PowerEdge 2650 servers, and we are now achieving 7.2 Gb/s with a 1.5-GHz Itanium-II system. Taking these same Intel cards and housing one at Sunnyvale, California and one at Geneva, Switzerland, we managed to smash the Internet2 Land Speed Record by a factor of 2.5, a record that had just been set three months prior. The long-range goal is to have our optimizations folded back into the Intel network interface cards so that application users can achieve such speeds directly out of the box (assuming that their network infrastructures can support such speeds).

4. Recent Accomplishments

Dynamic Right-Sizing (DRS)

- Completed the research, development, and alpha prototyping of a user-space implementation of DRS called drsFTP. Third-party transfers have not been implemented (yet).
- Released pre-alpha versions of the software to the NSF-funded optIPuter project (UCSD/SDSC and U. Illinois at Chicago) as well as the U.S. Air Force for enabling faster image processing.
- Completed testing of the Linux 2.4.x DRS kernel patch, including transparency issues where one end host may not be "DRS-ed."

- Disseminated an alpha release of DRS to the research community, particularly the grid community. (Note: The NSF-funded Web100 and DOE-funded Net100 projects already have a pre-alpha release of DRS that they have successfully integrated and have done preliminary testing with.)
- Implemented a pre-alpha prototype of DRS in GridFTP (prior to its XIO re-design).
- Completed a pre-alpha prototype of DRS in FreeBSD.

Rude TCP

- Completed the research and development of a pre-alpha prototype of Rude TCP in Linux.
- Tested Rude TCP in an isolated SAN-like environment and observed a 30% increase in throughput.

10-Gigabit Ethernet

- Completed optimizations necessary to achieve bandwidth and latency numbers that are competitive with (and in some cases, significantly better than) more exotic technologies such as Myrinet.
- Sustained 2.38 Gb/s for over an hour across a WAN between Sunnyvale, California and Geneva, Switzerland that had a 2.45-Gb/s OC-48 transoceanic bottleneck link, i.e., terabyte in an hour.

5. Future Plans

Dynamic Right-Sizing (DRS)

- Disseminate a beta release of DRS for Linux 2.4.x
- Port DRS to FreeBSD 3.1 and Linux 2.6.
- Complete testing of the drsFTP prototype with two-party transfers.
- Implement third-party transfers in drsFTP and then package and alpha release the software.
- Follow-up on case study of DRS by NSF Web100 and DOE Net100 projects.
- Follow-up on user case studies of drsFTP with the UCSD/SDSC and the U.S. Air Force.
- Complete the research, development, and deployment of an alpha release of DRS in the re-designed GridFTP / XIO architecture.
- Integrate DRS with FAST TCP from Caltech.

Rude TCP

- Complete testing of Rude TCP in both dedicated and shared network environments and then package and alpha release the software.

10-Gigabit Ethernet

- Continue working with Intel towards achieving a true 10 Gb/s from end-to-end.
- Push for getting a network processor on-board the network interface card.

6. Publications

W. Feng, G. Hurwitz, H. Newman, S. Ravot, R. 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," To appear in SC 2003, November 2003.

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

M. Gardner, S. Thulasidasan, and W. Feng, "User-Space Auto-Tuning for TCP Flow Control in Computational Grids," *Computer Communications*, 2003.

S. Thulasidasan, W. Feng, and M. Gardner, "Optimizing GridFTP Through Dynamic Right-Sizing," *IEEE Symposium on High-Performance Distributed Computing (HPDC'03)*, June 2003.

A. Feng, A. Kapadia, W. Feng, and G. Belford, "Packet Spacing: An Enabling Mechanism for the Delivery of Multimedia Content," *The Journal of Supercomputing*, Aug. 2002.

M. Gardner, W. Feng, and M. Fisk, "Dynamic Right-Sizing in FTP (drsFTP): An Automated Technique for Enhancing Grid Performance," *IEEE Int'l Symp. on High-Performance Distributed Computing*, Jul. 2002.

E. Weigle and W. Feng, "A Comparison of TCP Automatic-Tuning Techniques for Distributed Computing," *Proc. of the IEEE Int'l Symp. on High-Performance Distributed Computing*, Jul. 2002.
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*, 2002.
M. Fisk and W. Feng, "Dynamic Right-Sizing: TCP Flow-Control Adaptation" (Poster), *SC 2001*, Nov. 2001.
E. Weigle and W. Feng, "Dynamic Right-Sizing: A Simulation Study," *IEEE Int'l Conf. on Computer Communications & Networks*, Oct. 2001.

7. Invited Talk

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.

8. Media Coverage

"Los Alamos Sets Internet Speed Mark in Guinness Book," *GRIDtoday*, Vol. 2, No. 31, August 4, 2003. http://www.gridtoday.com/03/0804/101764.html.
"Internet Speed Record in Guinness World Records Book," *HPCwire*, July 25, 2003. http://www.tgc.com/hpcwire/hpcwireWWW/03/0725/105593.html.
"Los Alamos TCP Pipe Hits 8.5 Gbps," *SpaceDaily*, July 25, 2003. http://www.spacedaily.com/news/internet-03u.html.
"Los Alamos Sets Internet Speed Mark in Guinness Book," *GRIDtoday: Breaking News*, July 24, 2003. http://www.gridtoday.com/breaking/781.html.
"Internet Speed Record in Guinness Word Records Book," *EurekAlert!*, July 24, 2003. http://www.eurekalert.org/features/doe/2003-07/danl-ism072503.php.
"Data Speed Record Crushed," *The Register*, June 6, 2003. http://www.theregister.com/content/5/31085.html.
"Researchers Set Data Speed Record from U.S. to Europe," *InfoWorld*, March 17, 2003. http://www.infoworld.com/article/03/03/17/HNdataspeed_1.html.

9. References

[Allc01] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, S. Tuecke, "GridFTP: Protocol Extensions to FTP for the Grid," RFC Draft, August 2001.
[Beck02] M. Beck, T. Moore, J. Plank, "An End-to-End Approach to Globally Scalable Network Storage," ACM SIGCOMM 2002, August 2002.
[Duni02] T. Dunigan, M. Mathis, B. Tierney, "A TCP Tuning Daemon," SC 2002, November 2002.
[Feng03] W. Feng, G. Hurwitz, H. Newman, S. Ravot, R. Cottrell, O. Martin, F. Coccetti, C. Jin, D. Wei, S. Low, "Optimizing 10-Gigabit Ethernet for Networks of Workstations, Clusters, and Grids: A Case Study," To appear in SC 2003, November 2003.
[Gard03a] M. Gardner, M. Broxton, A. Engelhart, W. Feng, "MUSE: A Software Oscilloscope for Clusters and Grids," IEEE International Parallel & Distributed Processing Symposium (IPDPS'03), April 2003.
[Gard03b] M. Gardner, W. Feng, M. Broxton, G. Hurwitz, A. Engelhart, "Online Monitoring of Computing Systems with MAGNET," IEEE/ACM Symposium on Cluster Computing and the Grid (CCGrid'03), May 2003.
[Hurw03] G. Hurwitz, W. Feng, "Initial End-to-End Performance Evaluation of 10-Gigabit Ethernet," IEEE Hot Interconnects: A Symposium on High-Performance Interconnects," August 2003.
[Jin03] C. Jin, D. Wei, S. Low, G. Buhrmaster, J. Bunn, D. Choe, R. Cottrell, J. Doyle, H. Newman, F. Paganini, S. Ravot, S. Singh, "FAST Kernel: Background, Theory, and Experimental Results," International Workshop on Protocols for Fast Long-Distance Networks," February 2003.
[Petr02] F. Petrini, W. Feng, A. Hoisie, S. Coll, E. Frachtenberg, "The Quadrics Network (QsNet): High-Performance Clustering Technology" (Extended Version), *IEEE Micro*, January-February 2002.
[Ribe00] V. Ribeiro, R. Riedi, M. Crouse, R. Baranuik, "Multiscale Queueing Analysis of Long-Range-Dependent Network Traffic," INFOCOM'00, March 2000.
[Web100] http://www.web100.org.