Transport Requirements for High Performance Network Applications that are *NOT* FTP

J. Shalf, M. J. Bennett, and G. Bell
Lawrence Berkeley National Laboratory

The exponential growth of high bandwidth global networks has rapidly exposed the design limitations of the TCP protocol. The network research community has risen to the challenge of developing new protocols and methods that greatly improve the efficiency of bulk transfers over high bandwidth-delay product networks, but these solutions fail to address the needs of a wide variety of applications that do not involve long, sustained data transfers. These applications include metacomputing and task-farming codes that exhibit bursty high-throughput communication patterns during state synchronization; advanced remotely-accessible formats and distributed filesystems that exhibit small MTU-sized metadata requests mixed in with bulk data streams; highly interactive visualization applications with large instantaneous data-handling requirements; and of course high-definition, fixed data-rate network video streams. Current bulk-data-oriented solutions either fail to address these application requirements (as is the case for network sampling solutions), or they create the potential for rampant uncontrolled resource conflicts (as is the case for fixed rate solutions). In short, there is a clear need to redirect attention from bulk transfer requirements in order to determine how the current crop of proposed network protocols and network/grid infrastructure can be modified to enable an efficient, scalable solution to a wider variety of application requirements.

The congestion avoidance behavior of TCP has been implicated as a likely culprit for poor single-stream TCP throughput on high bandwidth-delay-product networks. The network research community has offered wide range of alternatives to circumvent the behavior of the aging AIMD congestion avoidance algorithm, including multistream/GridFTP, instrumented kernels such as Net100, delay-based congestion control with FAST, congestion hints from the switching infrastructure via XCP and EDN mechanisms, and non-AIMD congestion avoidance behaviors such as HSTCP. However, these new methods can exacerbate the problems experienced by applications which require bursty, high-throughput data flows, and which are intolerant of layer3/protocol-induced latency caused by excessive buffering. High bandwidth-delay product networks have a severe impact on the performance of these applications, but virtually no protocol solutions have addressed their requirements. Sample applications include:

- High-performance interactive visualization algorithms like IBR [1]
- Grid/metacomputing applications such as the Gordon Bell Award winning Cactus code [2,3]
- Complex remote data management applications like remote and streaming HDF5. [4]

For instance, interactive visualization applications typically require high-bandwidth, reliable network flows in order to minimize the apparent latency of a remote update from the user's perspective. It is not usually possible to pre-fetch data in order to conceal the

effect of latency, because data transfer is in direct response to manipulation of the user interface. However, since these flows are only periodic, it takes substantially longer to open up the congestion window so that the network's available capacity is fully utilized (even with faster-than-additive-rate-recovery). Furthermore, the TCP protocol must infer the available capacity of the network through continuous sampling of network conditions. However, when the application pauses the stream, the TCP protocol is rendered blind to the current congestive state of the network, resulting in potentially disruptive (and inappropriate) send-rates when the stream resumes. Finally, reliable protocols are tied inextricably to the enforcement of stream-semantics. Even with SACK, the receive host must buffer up to an entire TCP-window's-worth of data as it awaits the resend of missing packets that form gaps in the stream (with multiple windows' worth of buffering in the case of GridFTP). It seems clear that latency-sensitive applications require a reliable analog to the new high-speed protocols that can deliver reliable datagrams in addition to reliable streams – semantics offered by the emerging SOCK_RDM protocol, but not apparently with any consideration for congestion-avoidance requirements.

Consequently, many such applications have moved to fixed data-rate implementations that are already typical of network video and audio streams [1]. Fixed data-rate protocols ensure the best-possible utilization efficiency for explicitly allocated capacity, and can maintain these rates even when used in bursty fashion in competition with continuous streams. Few of these protocols provide SOCK_RDM-like semantics that are needed to reduce superfluous layer-3 buffering of data. But fixed-rate protocols may lead us to a situation similar to the 1990's, when fixed-rate network video streams were burning down the Internet. Existing QoS architectures, such as IntServ and DiffServ, offer tempting solutions for researchers interested in the problem of accommodating fixed-rate data streams, but have failed to garner widespread adoption due to their complexity and inter-domain coordination issues.[5] In practice, network managers almost always have found it more cost-effective to over-provision than to deploy QoS. The complexity and socio-political issues inherent in QoS technology will likely continue to impede progress on wide-area deployments, and yet we cannot continue to over-provision network infrastructure indefinitely to stave off resource conflicts because the new breed of fixed-rate protocols are fully capable of matching any such improvements. So there is a clear need to reevaluate the approach to resource management for fixed-rate streams.

It is critically important for the network community to engage in a discussion of the requirements of applications other than file transfer protocol! While it is possible that fixed rate protocols or switched lambdas can accommodate the needs of these applications, its important to evaluate the infrastructure required to move those methods out of the lab and into the shared production resources. Perhaps the answer is not in delivering complex QoS solutions that actively manage flows, but to provide simple feedback that enables those flows to make sensible decisions to self-limit their data rates at the endpoints.

References

1. J. Shalf and E. W. Bethel, "**Cactus and Visapult: An Ultra-High Performance Grid-Distributed Visualization Architecture Using Connectionless Protocols."** *IEEE Computer Graphics and Applications,* Volume 23, Number 2, March/April 2003.
2. Werner Benger, Ian Foster, Jason Novotny, Edward Seidel, John Shalf, Warren Smith and Paul Walker. "**Numerical Relativity in a Distributed Framework"** *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, San Antonio, TX. 1999.
3. Gabrielle Allen, David Angulo, Ian Foster, Gerd Lanfermann, Chang Liu, Thomas Radke, Ed Seidel**, "The Cactus Worm: Experiments with Dynamic Resource Discovery and Allocation in a Grid Environment,"** *International Journal of High Performance Computing Applications,* Volume 15, Number 4, Winter 2001.
4. Gabrielle Allen, Tom Goodale, Gerd Lanfermann, Thomas Radke, Edward Seidel, Werner Benger, Hans-Christian Hege, Andre Merzky, Joan Massó and John Shalf **"Solving Einstein's Equations on Supercomputers,"** *IEEE Computer,* Vol **32**, 1999.
5. Gregory Bell, "**Failure to Thrive: QoS and the Culture of Operational Networking**", *Proceedings of the ACM SIGCOMM 2003 Workshops, RIPQoS Workshop*, Karlsruhe, Germany, August 2003.