Improving the Bulk Data Transfer Experience

Chin Guok*, Jason R Lee, Karlo Berket
Lawrence Berkeley National Laboratory
1 Cyclotron Rd MS 50A-3125, Berkeley, CA 94720, USA
E-mail: chin@es.net
E-mail: jason@nersc.gov
E-Mail kberket@lbl.gov
*Corresponding author

Abstract: Scientific computations and collaborations increasingly rely on the network to provide high-speed data transfer, dissemination of results, access to instruments, support for computational steering, etc. The Energy Sciences Network is establishing a science data network to provide user driven bandwidth allocation. In a shared network environment, some reservations may not be granted due to the lack of available bandwidth on any single path. In many cases, the available bandwidth across multiple paths would be sufficient to grant the reservation. In this paper we investigate how to utilize the available bandwidth across multiple paths in the case of bulk data transfer.

Keywords: bulk data transfer; performance measurement; performance improvement; user driven bandwidth allocation; bandwidth reservation; traffic engineering; overlay networks; high-speed data transfer; science data network

Reference to this paper should be made as follows: Weng, T-h. and Chapman, B. (2004) ‘Towards optimisation of openMP codes for synchronisation and data reuse’, *Int. J. High Performance Computing and Networking*, Vol. 1, Nos. 1/2/3, pp.43–54.

Biographical notes: Chin Guok joined ESnet in 1997 as a Network Engineer, focusing primarily on network statistics. Chin received his Masters Degree in Computer Science from the University of Arizona. He is currently the principle investigator of the ESnet On-demand Secure Circuit and Advance Reservation System (OSCARS) project.

Jason Lee is a Network Engineer with the National Energy Research Scientific Computing Center (NERSC). Jason received his Masters Degree in Computer Science from SFSU. Jason has spent the last 15 years at LBNL working in network research. His current interests include 100G networking and provisioning network resources within a supercomputer center.

Dr. Karlo Berket is a Scientist in the Distributed Systems Department at the Lawrence Berkeley National Laboratory (LBNL). He received his Doctorate degree in Computer Engineering for his work on the InterGroup group communication system at the University of California, Santa Barbara. His current research interests include overlay networks and semantic management of metadata.

1 INTRODUCTION

The practice of modern science is increasingly dominated by large-scale collaborations of multi-disciplinary teams integrating results from both simulation and observation. Scientific computations and collaborations increasingly rely on the network to provide high-speed data transfer, dissemination of results, access to instruments, support for computational steering, etc. The data sets that need to be shared are increasingly reaching sizes in the terabyte (TB) range. This makes the function of the network increasingly critical to the success of such cooperative efforts. Recent news reports (Farivar, 2007; Waters, 2007) have picked up on the efforts to transfer the entire collection of Hubble telescope data (about 120 terabytes) to scientists at various research institutions by shipping hard disks via mail, because it’s faster than sending it over the network. Advances in storage and network technologies will help speed up bulk data transfers, but we can also improve performance by utilizing our current resources more wisely.

To address some of these issues, the Energy Sciences Network (ESnet) (Energy Sciences Network, 2007) is establishing a science data network that is logically separate from the production IP core network. This will provide the underlying capability required by scientific applications as identified in the 2002 U.S. Department of Energy (DOE) (Department of Energy, 2007) Office of Science workshop (High Performance Network Planning Workshop, 2002). One of the requirements of the science data network is the ability to provide user driven bandwidth allocation. The default characteristics of the Internet today do not provide a user any service guarantees. There is neither the assurance that a packet will be delivered to its destination, nor any transport predictability (such as latency and jitter) when the packet is in transit. The requirements for user driven bandwidth allocation has spawned several activities such as the DOE funded Lambda Station (Bobyshev et al., 2006), On-Demand Secure Circuits and Advance Reservation System (OSCARS) (Guok et al. 2006), TeraPaths (Bradley et al., 2006) and UltraScience Net (Rao et al., 2005) projects, the National Science Foundation (NSF) (National Science Foundation, 2007) funded Circuit-switched High-speed End-to-End Transport Architecture (CHEETAH) (Veeraraghavan et al., 2003) and Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON) (Yang et al., 2006) projects, Internet2's (Internet2, 2007) Bandwidth Reservation for User Work (BRUW) (Riddle, 2005) and Hybrid Optical and Packet Infrastructure (HOPI) (Boyles, 2004) projects, CANARIE's (CANARIE Inc., 2007) User-controlled Lightpath (UCLP) (Wu et al., 2005) project, and GÉANT's (GÉANT, 2007) AUTOBahn (Sevasti, 2006) and Advanced Multi-domain Provisioning System (AMPS) (Patil, 2006) activities.

The OSCARS proof-of-concept service has been deployed within the ESnet production network. OSCARS is designed as a service for dynamic QoS path establishment that is simple for users to use, and easy to administer. The user can make reservations either for immediate use or in advance for either one-time use or persistent use, e.g. for the same time everyday. The user does not have to configure an alternate routing path, nor mark the packets in any way. All necessary mechanisms needed to provide the user with a guaranteed bandwidth path are coordinated by a Reservation Manager (RM) and managed by the routers in the network. Traffic engineering is essential in making more efficient use of the network. In OSCARS, advanced users have the option to specify ingress and egress end-points (within ESnet) of the virtual circuit. This effectively allows the user to "route" around congested peering points if there are alternatives available. A near term enhancement to this is the ability for users to determine the path(s) the virtual circuit(s) will take as it traverses the ESnet backbone. This is executed via explicit Label Switched Paths (LSPs). To effectively reserve bandwidth in a network, which is a shared resource environment, appropriate authentication and authorization policies must be enforced to prevent abuse. Bandwidth on each link is allocated appropriately to prevent over-provisioning, and access controls must be implemented to prevent over-subscription.

In a shared network environment, some reservations may not be granted due to the lack of available bandwidth on any single path (e.g. 2Gbps reservation on a 1Gbps link). It may also be the case that a provider is unwilling to allocate a large portion of the bandwidth on a path to a single reservation. In many cases, the available bandwidth across multiple paths would be sufficient to grant the reservation. In this paper we investigate how to utilize the available bandwidth across multiple paths in the case of bulk data transfer. In particular, our goals are: to present a prototype implementation that enables multiple path allocation for a specific bulk data transfer protocol, GridFTP (Allcock et. al., 2005); to show that using multiple paths can improve the performance of the data transfer; and to show that using multiple paths can be used to improve the fairness of the network..

2 RELATED WORK

Provisioning guaranteed bandwidth paths in a network is not a novel idea. Protocols such as Multiprotocol Label Switching (MPLS) (Rosen, Viswanathan and Callon, 2001) and Reservation Protocol (RSVP) (Braden et. al., 1997) have provided network operators with this capability for some time. However, extending this functionality to an end user or application, which has little to no network traffic engineering knowledge, in a simple to use service is what makes this innovative. There are several systems deployed today in research and education networks that facilitate user driven guaranteed bandwidth provisioning. These generally fall into two categories; on demand provisioning (e.g. Lambda Station, CHEETAH, DRAGON, and HOPI) and advance reservation (e.g. OSCARS, BRUW, UCLP, AUTOBahn, and AMPS).

The area of TCP performance is rampant with literature on how to improve a single TCP stream. Contests such as the Land Speed Record (Internet2 Land Speed Record, 2007) put on by Internet2 fuel this desire to have the fastest (biggest) single stream of TCP data across the longest distance. This has led to a belief that there is no need for multiple stream TCP since the speed of a single stream is now so fast (average speed of 8.80 Gbps) that there seems little reason to try to use more then one stream as a single stream can now almost fill most backbone pipes (OC-192 and 10Gbps Ethernet). In practice, multiple stream TCP is still used in more instances since it still requires a lot of tuning to achieve this level of performance on a single stream. Also, not all backbones have the capacity of 10Gbps but instead are provisioned with multiple smaller links (OC-48, which is 2.5Gbps). This leads to situations where scientists may need to allocate (reserve and use) large amounts of bandwidth between sites, but are unable to allocate these resources because there is no single, sufficiently fast, path available.

There has been some work on splitting streams to multiple instances that can then be run across these smaller links. Heyman and Lucantoni, (2003) show that by using rate limiting to lower the effective bandwidth across links the aggregated bandwidth used can be maximized across all the streams (and links) to create a total bandwidth much larger than any single instance on these links. Experiments in how

TCP performs when split across multiple steams has been performed (Sivakumar, Bailey and Grossman, 2000) but these results haven't been scaled to study results across multiple, distinct, paths. Most tests suffer from the congestion caused by their own streams, while we are looking at the results obtained using separate paths.

3 IMPLEMENTATION

For the purposes of this paper, we call a unit of sequential code that will be executed by a single thread a task. Such a task is an arbitrary code sequence, such as a set of iterations of a loop nest, or one or more procedures, without synchronisation constructs. An OpenMP program will be decomposed into an ordered collection of tasks according to the semantics of the language. The ordering will be represented by the task graph. A task graph for an OpenMP program, denoted by $G(N, E)$ consists of a set of nodes $\boldsymbol{N} = \{t_1, t_2, \cdots, t_m\}$, where each node represents a task in the decomposed program, and a set of edges E between nodes, where $e_{i,j}$ is an edge from node t_i to node t_j. Each edge represents synchronisation in the sense that the source node must be executed before the sink node at run time. As part of our translation strategy, we will eliminate as many edges as possible from this graph in order to reduce the amount of synchronisation imposed on the executing code. We will also use this graph and our analysis to map the tasks in the program to threads. Note that we bind exactly one thread to each target processor so that we may refer to threads and processors interchangeably. We assume that processors (and threads) are numbered consecutively, beginning with one.

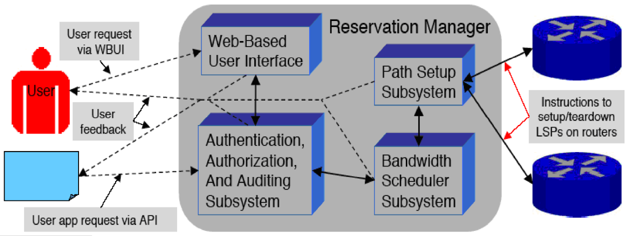
3.1 Generation of tasks and task graph

A system that provides bulk data transfer across multiple network paths can be considered as an instantiation of a virtual overlay network with the hosts and routers acting as peers. Instantiating virtual overlays can be decomposed into two tasks: translating a virtual overlay onto a physical network and routing application traffic onto the instantiated overlay.

In order to create the overlay network, the underlying network infrastructure must support two basic functions: management of circuits (i.e. setup, teardown) and enforcement of usage policies (i.e. AAA - authentication, authorization and accounting). We accomplish this by using the OSCARS system. The OSCARS system is comprised of three components: the Authentication, Authorization, and Auditing Subsystem (AAAS), the Bandwidth Scheduler Subsystem (BSS), and the Path Setup Subsystem (PSS) (Figure 1). Reservation request messages are passed using X.509 signed SOAP messages over SSL connections. This allows the authentication of both the client and the server. In addition, the signed SOAP message contains the X.509 certificate of the user. Using either or both the client and user certificates, usage policies are enforced accordingly using a role based lookup table. It must be noted that ESnet maintains the DOEgrids root CA, which simplifies the verification of certificates for the target users of this service.

In OSCARS, the traffic engineering aspects of circuit management is accomplished using Open Shortest Path First – Traffic Engineering (OSPF-TE) (Ishiguro et. al., 2007) to gain routing information, MPLS-TE (Rosen, Viswanathan and Callon, 2001) to enable switching, and RSVP-TE (Awduche et. al., 2001) as the signalling mechanism to provision the virtual circuits (LSPs). An added feature that OSCARS provides is the ability for a user to make a reservation for a future point in time. The network topology is stored in a database and the available bandwidth of each link is managed based on the various bandwidth requests.

Figure 1 OSCARS Architecture


User request via WBUI
User
User feedback
User app request via API
Reservation Manager
Web-Based User Interface
Path Setup Subsystem
Authentication, Authorization, And Auditing Subsystem
Bandwidth Scheduler Subsystem
Instructions to setup/teardown LSPs on routers

For reservations with multiple paths, all paths between the ingress and egress points are computed and only the “best” paths are selected and reserved in the database. The “best” paths are defined based on the users request parameters (e.g. maximum hops, latency, etc). Traffic rates for each path can be defined by the user in addition to path packet filtering based on IP flow-spec parameters such as port, or DSCP (Nichols et al., 1998) bits.

To enforce the bandwidth guarantees within the ESnet backbone, a separate QoS queue (expedited-forwarding (EF)) used exclusively for OSCARS circuits is configured to match the RSVP limits on a per interface bases (i.e. EF queue = max RSVP bandwidth = 50%). Unlike the other queues configured in the ESnet backbone, the EF queue is set to hard drop packets when the traffic rate exceeds the preset limit.

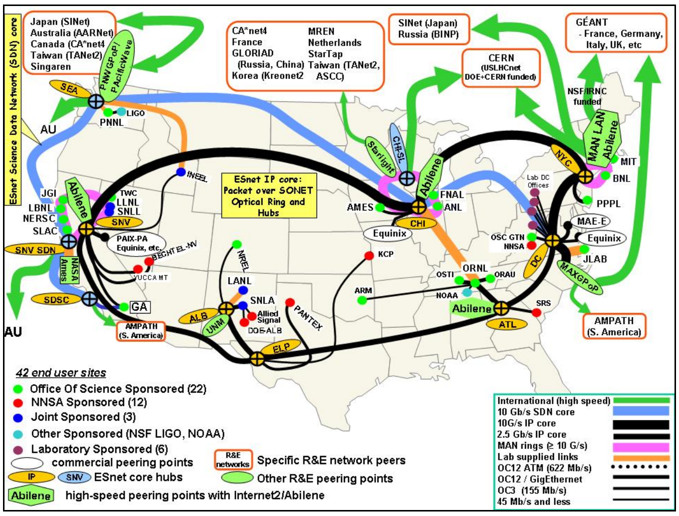
The central question in routing the application traffic onto the lower-layer overlay is at which layer, and at which granularity, does the application traffic get split into multiple flows at the source and rejoined at the sink. Possible methods include using a modified TCP stack to label application traffic at the source, offloading this functionality to an intermediate host or a programmable logic component, or varying source/destination port numbers and letting the router perform the tagging of the traffic. The resulting methods also need to deal with out-of-order packets, which can lead to buffering issues at both source and sink.

For the prototype implementation we chose GridFTP (Allcock et. al, 2005) as the bulk-data transfer protocol. GridFTP allows for the use of parallel streams so that the data transfer can be striped across several TCP streams simultaneously. This allows us to implement routing of the application traffic onto the lower-layer overlay by tagging the traffic according to source port numbers. When using parallel streams GridFTP runs in Extended Block Mode (MODE-E). This mode of operation supports out-of-order data delivery and allows our implementation to not have to deal with out-of-order packets.

4 EXPERIMENT SETUP

The network portion of this test consisted of setting up multiple diverse paths between a source host located in the Level 3 facility in Sunnyvale CA and a destination host located in the Starlight facility in Chicago IL. This was done over the ESnet production infrastructure (Figure 2). The test paths were set up using traffic engineered MPLS LSPs following the OSCARS methodology for configuring LSPs and traffic filtering. However due to the higher bandwidth requirements of the test, the LSPs were configured for best-effort service as oppose to expedited-forwarding, and no admission control was enforced.

Figure 2 ESnet production infrastructure (in early 2007)

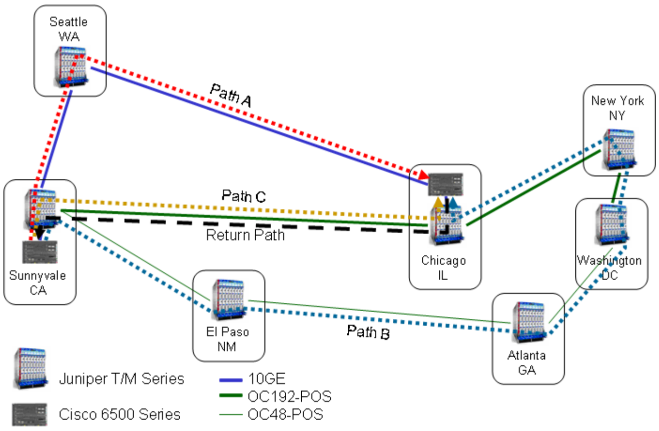


The test nodes were connected via three diverse unidirectional paths (Figure 3). The first path (A) traversed a 10Gbps (10GE) link from Sunnyvale CA up to Seattle WA, and then on to Chicago IL. The second path (B) was composed of 2.5Gbps (OC48-POS) and 10Gbps (10GE, OC192-POS) links from Sunnyvale CA, across to El Paso NM, and Atlanta GA, up to Washington DC, and New York NY, and then back to Chicago IL. The third path (C) took the most direct route over a 10Gbps (OC192-POS) from Sunnyvale CA, to Chicago IL. Table 1 shows the round trip latencies and maximum bandwidth for the paths. All return traffic from Chicago to Sunnyvale took the direct OC192-POS link.

Table 1 Experiment path characteristic

Path	Roundtrip Latency (ms)	Maximum Bandwidth (Gpbs)
A	57	10
B	75	2.5
C	48	10

Figure 3 Network test setup



The traffic selection for each of the three paths was based on the source port number (in conjunction with the source and destination IP address) (see Table 2). This function was implemented via a firewall filter on the Juniper router at Sunnyvale CA where the source host was located.

Table 2 Experiment paths and ports

Path	Source Ports
A	30010-30019, 30040-30042, 30050-30059
B	30020-30029, 30043-30045
C	30000-30009, 30030-30039, 30046-30049

Our end nodes were Dual 2.6 GHz AMD Opteron processors, each with 2 Gigabytes of memory. Each end node was directly connected to the network by 10Gbps network interfaces (Myricom, 2007). These hosts were running RedHat 4.1.1-51 with a 2.6.19 kernel with Binary Increase Congestion (BIC) TCP (Xu, Harfoush and Rhee, 2004). These hosts had been tested previous to our use and had shown sustained bandwidths of up to 7 Gbps TCP.

The testing was broken into runs. Each run was composed of 20 transfers. Each transfer moved 100GB of data from Sunnyvale to Chicago. We performed a run on each path and every combination of 2 paths.

For our testing we used the globus-url-copy application (globus-url-copy, 2007) to transfer the data using the GridFTP protocol. globus-url-copy has been tuned for use over Wide Area Networks by allowing for the independent tuning of both the sender and receiver TCP windows and the size of the writes to the network. We didn't modify globus-url-copy and used the latest released version (4.6). All of our transfers were done using '/dev/zero' as the input for the transfer and '/dev/null' as the output. This allows us to see that actual number that could be obtained if disk speeds were not an issue. Tuning a high-speed disk array for good disk performance is outside the scope of testing. The parameters and their values passed to globus-url-copy in our test are shown in Table 3.

Table 3 Parameters passed to globus-url-copy

Parameter	Value
Parallel	10
Block-size	1048576
Tcp-buffer-size	62500000

5 RESULTS

We performed the first set of experiments on the individual paths in order to get a baseline for the rest of the tests. We performed a run on each of the paths using 10 parallel streams for each transfer. The results, shown in Figure 4, indicate that we could only utilize around 55% of the capacity on paths A and C. Both of these paths have a 1.25GBps (10Gbps) capacity, but we were only able to reach sustained peak rates of 700MBps (5.6Gbps) on these links. This is due to a combination of a number of factors:

- End-host capability: In previous network testing, using Iperf (Iperf, 2005), the end hosts showed a 7 Gbps sustained transfer rate over path C. This shows that these end-hosts can only provide a 70% maximum utilization of the 10Gbps paths.
- GridFTP overhead: We believe that the addition of GridFTP headers and the read/write system calls account for the rest of utilization drop.

Conversely, we were able to utilize path B at over 80% of capacity. Path B has a 312MBps (2.5Gbps) capacity and we were able to reach sustained transfer rates of 260MBps (2.1 Gbps).

Figure 3 also shows that transfers on path C achieve a higher transfer rate than those on path A even though they have the same capacity. This is due to cross-traffic on path A. Path A is used as a production backup link, and is typically utilized for testing otherwise. At the time of testing, there were several other tests running concurrently that were independent from our tests.. We believe this was the primary cause of the lower average and higher jitter in the transfer rate measurements on this path.

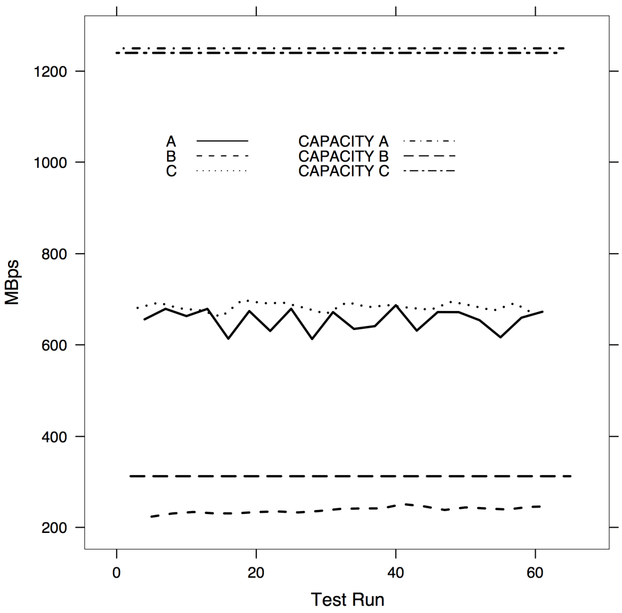
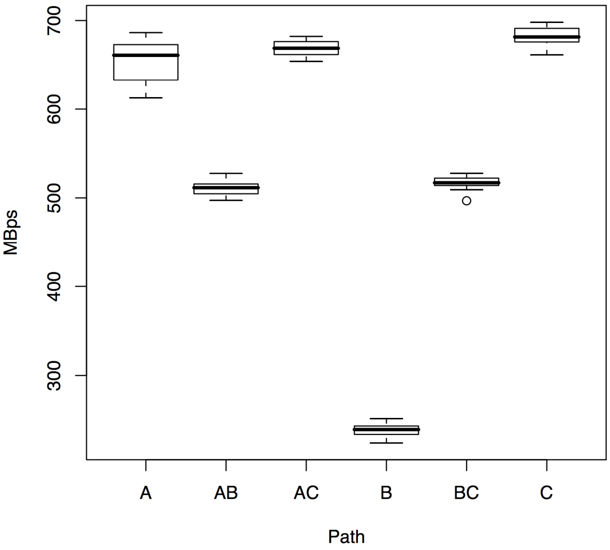


Figure 4 Data transfer rates on the three individual paths from Sunnyvale to Chicago (10 streams per path)

Next, we performed a run on each 2-path combination, i.e. AB, BC, and AC, splitting the number of streams evenly between the paths. For each path combination five streams were sent over each path. Figure 5 shows the average transfer rates for each path combination, as well as the baseline transfer rates. The results in Figure 4 were somewhat unexpected. The documentation of GridFTP (Allcock et. al, 2005) and the globus-url-copy program (globus-url-copy, 2007) suggests that the data to be transferred is dynamically allocated to the individual streams based on individual stream performance. In this case, the multi-path transfer rates would, ideally, be equal to the transfer rate of the faster path. By examining the rates for paths AB and BC, this is clearly not the case. The measured transfer rates indicate that globus-url-copy is sending half the transfer on each path, i.e. each stream is transferring 10% of the total data.

Figure 5 Data transfer rates on paths from Sunnyvale to Chicago (individual paths have 10 streams per path, two-path combinations have 5 streams per path)



For example let's look at the results for paths B, C, and BC. The measured average transfer rate for these paths is 238MBps, 682 MBps, and 517MBps. By dividing the amount of data sent by the data transfer rate, we get the average time per transfer:

$$time\ of\ \ transfer = \frac{amount\ of\ \ data}{transfer\ rate}$$

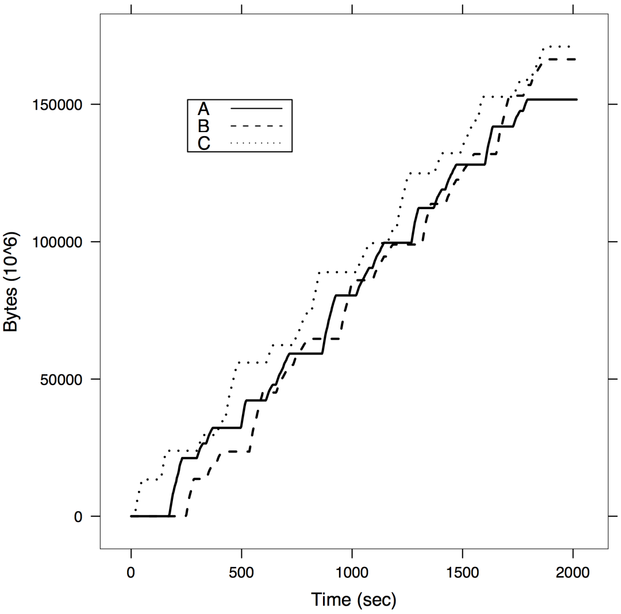
The average time per transfer for paths B, C, and BC is 420.1 sec, 146.6 sec, and 193.4 sec. If the data is being evenly distributed over all of the streams, then multi-path experiment BC will send half of the data on path B and half the data on path C and the time to complete the transfer on BC will be the maximum of the times to complete half the transfer on the individual paths:

$$time\ to\ transfer\ (ttt)\, 100GB\ on\ \ path\ BC = \max\left[ttt\ 50GB\ on\ \ path\ B,\, ttt\ 50GB\ on\ \ path\ C\right] = \max\left[210\sec,\, 73.3\sec\right] = 210\sec$$

and a transfer speed of 476MB/sec (100GB /210 sec).

During this run, we maintained byte counters on each of the paths. Figure 6 shows the number of bytes transferred across each path over time. It is clear that approximately the same number of bytes was transferred across each of the paths during the run.

Figure 6 Number of bytes transferred across each physical path



Based on the above observation it was determined that GridFTP did not dynamically allocated data to the individual streams based on the stream's performance but allocated the data evenly across the total number of streams.

The measured and expected average data transfer rates for all of the paths are shown in Table 4.

Table 4 Average data transfer rates and times.

	Path A	Path B	Path C	Path AB	Path AC	Path BC
Measured average data transfer rate (MB/sec)	654	238	682	510	669	517
Average time per 100GB transfer (sec)	152.8	420.1	146.6	195.9	149.6	193.4
Estimated time per 50GB transfer	76.4	210	73.3	97.9	74.8	N/A
Estimated transfer rate for 5 streams per path (MB/sec)	N/A	N/A	N/A	N/A	N/A	N/A

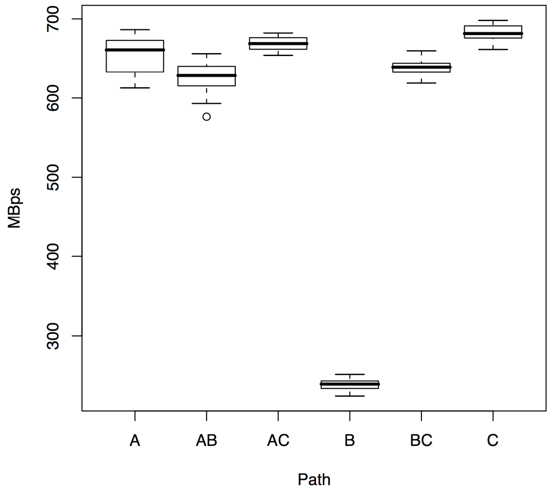
Next, we constructed a formula to determine the percentage of data to send over each of the paths in a multi-path transfer involving two paths to achieve the transfer rate of the faster path.

$$\begin{gathered}
\textit{time to transfer}(\textit{path } x) = \textit{time to transfer}(\textit{path } y)\\
\textit{data}(\textit{path } x) \times \textit{transfer rate}(\textit{path } x) = \textit{data}(\textit{path } y) \times \textit{transfer rate}(\textit{path } y)\\
\frac{\textit{data}(\textit{path } x)}{\textit{data}(\textit{path } y)} = \frac{\textit{transfer rate}(\textit{path } x)}{\textit{transfer rate}(\textit{path } y)}
\end{gathered}$$

Note that this formula may give optimistic values and that the maximum transfer rate of any bottleneck needs to be taken into account.

Using this formula we see that optimum split for the multi-paths in our experiment that involve path B is to send 35% of the traffic on path B and 65% of the traffic on the other path (A or C). Since we use 10 streams in total for each transfer, having 3 streams (and thus 30% of the traffic) on path B and 7 streams (70% of the traffic) on the other path. Figure 7 shows the results of adjusting the number of streams for transfer on AB and BC to this ratio. We can see that the transfer rate on the combined paths is now approximately the same as the transfer rate on the faster path. We still have a little bit of performance decrease on the multi-path. Further investigation is necessary to determine the cause for this.

Figure 7 Data transfer rates on paths from Sunnyvale to Chicago (individual paths have 10 streams per path, two-path combinations have 3 streams on the lower capacity path and 7 streams on the higher capacity path)



The results in Figure 7 show that by routing a bulk data transfer over multiple paths we can provide similar transfer rates to those attained by only using the higher performance path. In addition, using multiple paths alleviates the load on a single path. In this particular experiment we achieved comparable performance while sending 70% of the data on the high performance path.

6 CONCLUSION AND FUTURE WORK

In this paper we've investigated how to utilize the available bandwidth across multiple paths in the case of bulk data transfer. In particular, we've presented a prototype implementation that enables multiple path allocation for a specific bulk data transfer protocol, GridFTP. We've used this implementation to show that using multiple paths can match the performance of bulk data transfer over a single path and to show that using multiple paths can be used to improve the fairness of the network. We have not shown that using multiple paths can improve the performance of bulk data. We are currently setting up an experiment to demonstrate this.

Future work possibilities include: (1) implementing prototypes of this service for other applications (other parallel port ones should just work; possibly more generic service); (2) implementing a production version of this service for GridFTP; (3) algorithms for finding and providing reservations across multiple paths; and (4) changing paths mid-transfer (if can't reserve same paths for the duration of the transfer).

7 ACKNOWLEDGEMENT

The Energy Sciences Network (ESnet) is a high-speed network serving tens of thousands of Department of Energy scientists and collaborators worldwide. ESnet provides high bandwidth network connections to all major DOE sites and, via comprehensive connections to the world's research and education (R&E) networks, to essentially all U.S. and international R&E institutions in order to support the science mission of the DOE Office of Science (SC), as well as several other DOE Offices. Esnet is also a tier 1 ISP with direct peerings with all major commercial networks.

REFERENCES

Allcock, W., Bresnahan, J., Kettimuthu, R., Link, M., Dumitrescu, C., Raicu, I. and Foster, I. (2005) 'The Globus Striped GridFTP Framework and Server', *Proceedings of Super Computing 2005 (SC05)*, Seattle, WA, November, pp.54.

Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V. and Swallow, G. (2001) *RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels*, [Online] Available at: ftp://ftp.rfc-editor.org/in-notes/rfc3209.txt [13 April 2007]

Bobyshev, A., Crawford, M., DeMar, P., Grigaliuna, V., Grigoriev, M., Moibenko, A., Petravick, D., Rechenmacher, R., Newman, H., Bunn, J., van Lingen, F., Nae, D., Ravot, S., Steenberg, C., Su, X., Thomas, M. and Xia, Y. (2006) 'Lambda Station: Production Applications Exploiting

Advanced Networks in Data Intensive High Energy Physics', *Computing in High Energy and Nuclear Physics (CHEP),* Mumbai, India, [Online] Available at: http://indico.cern.ch/getFile.py/access?contribId=162&sessionId=6&resId=0&materialId=paper&confId=048 [5 May 2008].

Boyles, H. (2004) 'Recent Results from Internet2's Hybrid Optical and Packet Infrastructure Project (HOPI)', *TERENA Networking Conference*, Rhodes, Greece, [Online] Available at: http://tnc2004.terena.org/core_getfile.php?file_id=401 [5 May 2008].

Braden, R., Zhang, L., Berson, S., Herzog, S. and Jamin, S. (1997) *RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*, [Online] Available at: ftp://ftp.isi.edu/in-notes/rfc2205.txt [13 April 2007]

Bradley, S., Burstein, F., Cottrell, L., Gibbard, B., Katramatos, D., Li, Y., McKee, S., Pope-scu, R., Stampf, D. and Yu,D. (2006) 'TeraPaths: A QoS-enabled collaborative data sharing infra-structure for peta-scale computing research', *Computing in High Energy and Nuclear Physics (CHEP) Mumbai*, India, [Online] Available at: http://indico.cern.ch/getFile.py/access?contribId=162&sessionId=6&resId=0&materialId=paper&confId=048 [5 May 2008].

CANARIE Inc. (2007) [Online] Available at: http://www.canarie.ca/ [13 April 2007]

Department of Energy (2007) [Online] Available at: http://www.energy.gov/ [13 April 2007]

Energy Sciences Network (2007) [Online] Available at: http://www.es.net/ [13 April 2007]

Farivar, C. (2007) "Google's Next-Gen of Sneakernet", *Wired Magazine*, [Online] Available at: http://www.wired.com/science/discoveries/news/2007/03/73007 [12 April 2007]

GÉANT (2007) [Online] Available at: http://www.geant.net/ [13 April 2007]

Globus-url-copy (2007) [Online] Available at: http://www.globus.org/toolkit/docs/4.0/data/gridftp/rn01re01.html [13 April 2007]

Guok, C., Robertson, D., Thompson, M., Lee, J., Tierney, B. and Johnston, W. (2006) 'Intra and Interdoman Circuit Provisioning Using the OSCARS Reservation System', *International Conference on Broadband Networks (BroadNets 2006)*, San Jose, CA, pp.1-8.

Heyman, D.P. and Lucantoni, D. (2003) 'Modeling multiple IP traffic streams with rate limits', *IEEE/ACM Transactions on Networking*, Vol. 11, No, 6, pp.948- 958.

High Performance Network Planning Workshop (2002) [Online] Available at: http://www.doecollaboratory.org/meetings/hpnpw/index.html [13 April 2007]

Internet2 (2007) [Online] Available at: http://www.internet2.edu/ [13 April 2007]

Internet2 Land Speed Record (2007) [Online] Available at: http://www.internet2.edu/lsr/ [12 April 2007]

Iperf (2005) [Online] Available at: http://dast.nlanr.net/Projects/Iperf/ [13 April 2007]

Ishiguro, K., Manral, V., Davey, A. and Lindem, A. (Ed.) (2007) "Traffic engineering extensions to OSPF version 3", IETF Internet Draft, [Online] Available at: http://tools.ietf.org/html/draft-ietf-ospf-ospfv3-traffic-08 [13 April 2007]

Myricom (2007) [Online] Available at: http://www.myri.com/ [13 April 2007]

National Science Foundation (2007) [Online] Available at: http://www.nsf.gov/ [13 April 2007]

Nichols, K., Blake, S., Baker, F. and Black, D. (1998) *RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, [Online] Available at: ftp://ftp.rfc-editor.org/in-notes/rfc2474.txt [13 April 2007]

Patil, A. (2006) 'Advance multi-domain provisioning system', *TERENA Networking Conference*, Catania, Italy, [Online] Available at: http://www.terena.org/events/tnc2006/core/getfile.php?file_id=760 [5 May 2008].

Rao, N.S.V., Wing, W.R., Carter, S.M. and Wu, Q. (2005) 'Ultra-Science Net: Network testbed for large-scale science applications', *IEEE Communications Magazine*, Vol. 43, no. 11, pp.S12-17.

Riddle, B. (2005) 'BRUW: A bandwidth reservation system to support end-user work', *TERENA Networking Conference*, Poznan, Poland, [Online] http://tnc2005.terena.org/core/getfile.php?file_id=253 [5 May 2008].

Rosen, E., Viswanathan, A. and Callon, R. (2001) *RFC3031: Multiprotocol Label Switching Architecture*, [Online] Available at: ftp://ftp.rfc-editor.org/in-notes/rfc3031.txt [13 April 2007]

Sevasti, A. (2006) 'GN2-JRA3: A multi-domain bandwidth on demand service for the NREN community', *TERENA Networking Conference*, Catania, Italy, [Online] http://www.terena.org/events/tnc2006/core/getfile.php?file_id=1154 [5 May 2008].

Sivakumar, H., Bailey, S. and Grossman, R.L. (2000) 'PSockets: The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks', *SC2000: High-Performance Network and Computing Conference*, Dallas, TX, article no. 37.

Veeraraghavan, M., Zheng, X., Lee, H., Gardner, M. and Feng, W. (2003) 'CHEETAH: Circuit-switched high-speed end-to-end transport architecture', *Proceedings OptiComm 2003*, Dallas, TX, Vol. 5285, pp.214-225

Waters, D. (2007) "Google helps terabyte data swaps", BBC News, [Online] Available at: http://news.bbc.co.uk/2/hi/technology/6425975.stm [12 April 2007]

Wu, J., Savoie, M., Campbell, S., Zhang, H., Bochmann, G.V. and St. Arnaud, B. (2005) "Customer-managed end-to-end lightpath provisioning," International Journal of Network Management, Vol. 15, No. 5, pp.349-362.

Xu, L., Harfoush, K and Rhee, I. (2004) "Binary increase congestion control (BIC) for fast long-distance networks." INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. Hong Kong, March 7-11, Vol. 4, pp.2514-2524.

Yang, X., Lehman, T., Tracy, C., Sobieski, J., Gong, S., Torab, P. and Jabbari, B., (2006) "Policy-based resource management and service provisioning in GMPLS networks," in First IEEE Workshop on "Adaptive Policy-based Management in Network Management and Control", at IEEE INFOCOM 2006, Barcelona, Spain, pp.1-12.