

Exploiting Network Parallelism for Improving Data Transfer Performance

Dan Gunter*, Raj Kettimuthu†, Ezra Kissel‡, Martin Swany‡, Jun Yi§, Jason Zurawski¶

* Advanced Computing for Science Department, Lawrence Berkeley National Laboratory, Berkeley, CA

† Mathematics and Computer Science Division, Argonne National Laboratory Argonne, IL

‡ School of Informatics and Computing, Indiana University, Bloomington, IN

§ Computation Institute, University of Chicago/Argonne National Laboratory Chicago, IL

¶ Internet2, Washington, DC

Abstract—Many scientific applications, including bulk data transfer, can achieve significantly higher performance from virtually loss-free dedicated resources provisioned on shared links than from opportunistic network use. Research and Education (R&E) backbones, including the Energy Sciences Network and Internet2, provide general-purpose services to allocate dedicated bandwidth. However, in order to fully take advantage of this technology, applications need to move from coarse-grained “reservation” strategies, to more sophisticated control based on software defined networking (SDN) with technologies such as OpenFlow. We propose here, as one practical step in this direction, using multiple paths for the same application transfer session. This can add bandwidth from “best effort” and dedicated networks, and can also facilitate performance with applications using multiple 10G NICs over 100G capable paths.

I. INTRODUCTION

Enormous progress has been made in networking, to support scientific use cases. Research and Education (R&E) backbones, including the Energy Sciences Network and Internet2, provide general-purpose services to allocate dedicated bandwidth resources. The concept of “bandwidth reservation” introduces scheduling concerns, due to the relatively coarse grain nature of the allocations and inevitable fragmentation of the available capacity in a complex system such as multi-path network. Thus, links may appear full when in reality there may be multiple independent paths with aggregate available bandwidth to satisfy the request. With the growth of SDN technologies such as OpenFlow [19], applications may exert finer-grained control over network control strategies. Commercial entities are beginning to actively explore this space, e.g., Google is using OpenFlow on custom-designed hardware for all the internal networks it runs connecting its global data centers.

It is possible for applications to improve their overall performance expectations by using a combination of traditional “best effort” services and dynamic resource allocation with, e.g., OSCARS [20] or Software Defined Networking (SDN). To meet this challenge, we propose the use of multiple paths for the same application transfer session. This may also facilitate performance with applications using multiple 10G NICs over 100G capable paths.

GridFTP is widely used for bulk data movement, and serves as the basis for many “intelligent” applications including

PhEDEx [3] and Globus Online [1]. GridFTP transfers frequently make use of parallel TCP streams, often along the same network topology. Intelligent session layer “inlays”, such as the Phoebus [18] and XSP [17] infrastructure, enable access to WAN acceleration techniques and transparent use of dynamic networking technologies. To provide feedback from the infrastructure (both local to the machine and from the network at large), embedded NetLogger [24] and perfSONAR [12] monitoring can detect whether disks or networks are the limiting segment.

The SCinet Research Sandbox will allow us to evaluate strategies that transparently map traffic to different paths, using available SDN capabilities. We will monitor the performance using our monitoring architecture, *Periscope*, and dynamically adapt the use of multi-path capabilities based on knowledge of the end-to-end bottleneck.

II. BACKGROUND

The proposed experimentation will use a combination of state-of-the-art tools for wide area data movement, dynamic network adaptation, cross-domain bandwidth reservation, and end-to-end monitoring. This section introduces the relevant technologies for each of these tasks.

A. Wide-area Data Movement: GridFTP and Globus Transfer

The GridFTP protocol specification [4] extends the File Transfer Protocol (FTP) to provide secure, reliable, and efficient transfer of data across wide-area distributed networks. GridFTP extensions provide for, among other things, *striping* (i.e., the use of multiple data movers on a cluster file system), *parallelism* (i.e., the use of multiple socket connections between pairs of data movers), restart markers, and data channel security. Globus GridFTP [6] is an open source GridFTP implementation developed primarily at Argonne National Laboratory and the University of Chicago. The pluggable architecture allows for easy substitution of alternative authentication, authorization, transport, and file system access methods. In particular, the eXtensible I/O (XIO) interface [5] permits the substitution of alternative transport protocols and alternative file system interfaces.

Wide-area file transfer involves optimal use of a variety of complex resources and protocols. This task is inherently

complicated; GridFTP and similar tools have historically been complicated for the end-user to control and to fully understand. In addition, data transfers of large size may take an extended time period to complete. This introduces reliability and recovery problems that the user should not have to deal with directly. To address these problems, a Software-as-a-Service (SaaS) solution to manage transfers, now called Globus Transfer, was conceived and a prototype developed under the DOE CEDPS SciDAC project. The first production-level Globus Transfer system, delivered in November 2010, implements methods for managing the transfer of single files, sets of files, and directories, as well as *rsync-like* directory synchronization. It can manage security credentials, including for transfers across multiple security domains; select transfer protocol parameters for high performance; monitor and retry transfers when there are faults; and allows the user to monitor status.

B. Wide Area Network Control: Internet2 ION, NDDI, and OS³E

In [23], the authors made the argument that all of the intelligence of a network exists at the “edge”, while the capability of the “intermediary” devices in the middle should not be relied on beyond simple activities. Intelligence within the network now abounds due to advances in the concepts of Software Defined Networking (SDN), including approaches such as **OSCARS** and **OpenFlow**. Internet2 has embarked on two such offerings that allow control of the underlying networking by applications positioned at higher layers of the OSI stack.

1) *Internet2 ION*: Internet2 ION is an Interoperable On-demand Network service that affords users the ability to provision dedicated circuits (e.g. Layer 2 VLANs) across the Internet2 Network and affiliated partners. Users can reserve in advance, or provision in real time, point-to-point virtual circuits with whatever capacity they need to meet an application’s requirements. Built on widely deployed protocols, Internet2 ION can instantly reserve and create circuits across the Internet2 Network and into partner regional, national and international networks to connect researchers with colleagues worldwide [13].

2) *NDDI and OS³E*: Internet2, Indiana University and Stanford University have formed the Network Development and Deployment Initiative (NDDI) partnership, to create a new nationwide network platform to support global scientific research. A new Internet2 Network service, called the Open Science, Scholarship and Services Exchange (*OS³E*), will be offered on top of this network, and will be constructed using the first production deployment of OpenFlow technology. NDDI will deliver SDN, allowing researchers to experiment with new Internet protocols and architectures, and at the same time enabling domain scientists to accelerate their research with collaborators worldwide [2].

C. Dynamic Network Adaptation: XSP and Phoebus

The eXtensible Session Protocol (XSP) [17] has been designed as a flexible protocol architecture to manage the interactions of applications and network-based services, as

well as the devices that provide those services. Residing in layer-5 of the OSI network model [10], the notion of a *session* provides a natural scope for state, including establishment and lifetime of connections, forwarding rules, data transfer context or any other configuration related to the activity defined by the given interaction. Applications can refer back to that state, make changes, and signal the network to suit their particular requirements.

Most relevant to this work, XSP provides mechanisms that enable the configuration of dynamic networks services in support of applications such as GridFTP. The XSP libraries and API provide a standard interface for applications to specify parameters that define network paths. The realization of these paths is then managed by our XSP *Daemon* (XSPd) that signals the underlying provisioning service while providing feedback to the application. XSPd contains a number of modular backends for configuring devices that speak a particular protocol implementation, e.g., OpenFlow, OSCARS, Terapaths [11], etc. A transparent XSP wrapper, or “shim”, library gives existing application the ability to signal XSPd without source code modifications.

A key capability enabled via XSP is the notion of application-driven SDNs. In this model, applications are able to take a more active and direct role in configuring the underlying network in support of their data movement requirements. This may involve specifying a complete end-to-end path based on exchanged topology information, or simply signaling XSPd that a new transfer has begun while leaving the details of resource provisioning to local end-site policy. Within the XIO framework, we have implemented an XSP transform driver, *xio-xsp*, that gives Globus applications such as GridFTP the ability to enable XSP-specific features in conjunction with other drivers on the stack. When enabled, the driver effectively makes the GridFTP application “XSP-aware”, and imparts the ability to establish sessions to remote session peers. The *xio-xsp* driver allows GridFTP clients to indicate path configuration parameters, specify an XSPd instance that will realize those paths, and perform active signaling to notify other XSP services (e.g., Periscope monitoring) of transfer start/end events. The proposed experiment will investigate the role of XSP, and *xio-xsp* in particular, in provisioning OSCARS circuits and manipulating the SRS OpenFlow network, as well as enabling our end-to-end monitoring components.

Phoebus [18] is a performance-enhancing network service based on the XSP protocol framework. Designed to improve end-to-end throughput for long-distance data transfers, Phoebus embeds greater intelligence within the network, allowing it to choose the best transport technology based on application requirements and the available network resources at application run-time. Phoebus works by transparently splitting the full network path into distinct segments at specific adaptation points called Phoebus Gateways (PGs) and then finding the best path for the data on a segment-by-segment basis. These sub paths could include a combination of IP and dedicated optical circuits. Phoebus can also take advantage of dedicated circuits provisioned across backbone networks (e.g., between PGs) by providing an “on-ramp” for application flows that would otherwise be unable to make use of these advanced net-

works. With little to no modification needed by the end-user, applications that utilize Phoebus have experienced significantly improved throughput [21].

D. Bandwidth reservation: DYNES Project

Scientific innovation continues to increase requirements for the computing and networking infrastructures of the world. Collaborative partners, instrumentation, storage, and processing facilities are often geographically and topologically separated, thus complicating the problem of end-to-end data management. In addition to overall network design, the adoption of advanced network services has had a pivotal role in supporting both enterprise and data intensive science requirements. Many of the aforementioned technologies (*OSCARS*, *perfSONAR*, etc.) were originally targeted toward the largest component of the worldwide R&E networking infrastructure: Backbone and Regional providers. These far reaching networks, designed to offer large amounts of available capacity, may become congested in certain geographical areas that experience high demand. The ability to create a “protected” path, separate from the general-purpose IP traffic and using existing network resources, remains highly desirable for bulk data movement applications; particularly those relying on the TCP and operating on high RTT and high bandwidth paths.

The DYNAMIC NETWORK SYSTEM (DYNES) addresses this technology gap by providing hardware and software solutions to regional and campus networks - the overarching goal being to extend the technology already available on many backbone networks. This NSF sponsored project is developing and deploying a nationwide “cyber-instrument”, designed to span approximately 40 US universities and 11 regional networks. DYNES was awarded to a collaborative team including Internet2, the California Institute of Technology, the University of Michigan, and Vanderbilt University in 2010.

By integrating existing and emerging protocols, software for dynamic circuit provisioning and scheduling, in-depth end-to-end network path and end-system monitoring, and higher level services for management on a national scale, DYNES will allocate and schedule channels with bandwidth guarantees to several classes of prioritized data flows with known bandwidth requirements, and to the largest high priority data flows, enabling scientists to utilize and share network resources effectively.

E. End-to-end Monitoring: Periscope

We will drive our approach for the use of dynamic networks from real-time fine grained measurements collected via the Periscope system, which is a monitoring framework that provides an end-to-end view of system performance to a user. Periscope, based on technologies innovated by **perfSONAR**, presents data from the processes, hosts and various relevant points in the network in a single, scalable analysis and presentation framework. The context of a users activity is captured by the use of our XSP session protocol that initiates and controls various monitoring activities. In addition, Periscope uses this context to gather performance information from ongoing measurement activities on and between relevant

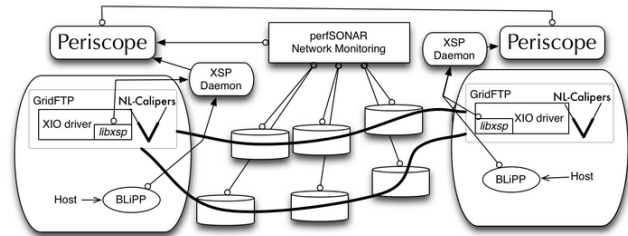


Fig. 1: Periscope architecture diagram.

network elements. The role of Periscope in the context of a measurement collection architecture is shown in Figure 1.

Performance information is provided to Periscope by our adaptive host-level probes and application instrumentation. The Basic Lightweight PerfSONAR Probes (BLiPP) read values from the operating system as directed through configuration and signals from XSP. The integration of XSP into BLiPP enables it to adapt its measurement collection to dynamic system state. To gather detailed application-layer performance measures in real time, we have integrated into the *xio-xsp* driver a log summarization extension to NetLogger called NetLogger Calipers (NL-Calipers). NL-Calipers is a user-level C library that measures the duration of an I/O operation and associates that duration with the number of bytes read or written, the I/O resource, and the context for the session. The overhead of instrumenting every I/O operation is minimized by efficient in-memory summarization, which is periodically emitted using Periscope formats as XSP metadata. The forwarding and control of measurement data is managed by modules written for the XSP Daemon. XSPd enables the interaction, in terms of both control and measurement transport, between Periscope and the underlying measurement collectors such as the instrumented *xio-xsp* driver and BLiPP.

III. EXPERIMENT OVERVIEW

Our experimental goals are twofold:

- 1): Improve transfer performance for a key data movement tool, namely GridFTP, through the use of multi-path transfers.
- 2): Explore intelligent decision making about when and how the use of network parallelism is beneficial based on active end-to-end monitoring and analysis.

Our proposed experiments will take advantage of both the dedicated and best-effort paths available between DYNES and other R&E end-sites to which we can provision network resources. Within the SCinet SRS at SC12, we intend to emulate such an end-site and make use of the sandbox environment to enable and explore a number of multi-path transfer approaches. We will leverage GridFTP’s ability to stripe application data over parallel socket streams to provide multiple flows of data for a single transfer operation. Application-driven signaling of the network via *xio-xsp* will allow for the redirection of these flows based on the current capabilities of the network, which may include forwarding over Phoebus Gateways. Our final task involves using end-to-end measurement data to react to and influence the extent of multi-path behavior.

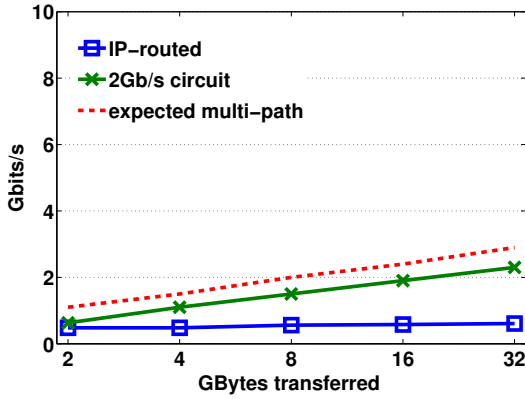


Fig. 2: Average performance for increasing transfer sizes between UMich and Caltech DYNES installations. 1Gb/s shared IP and 10Gb/s circuit interfaces using 8 parallel streams.

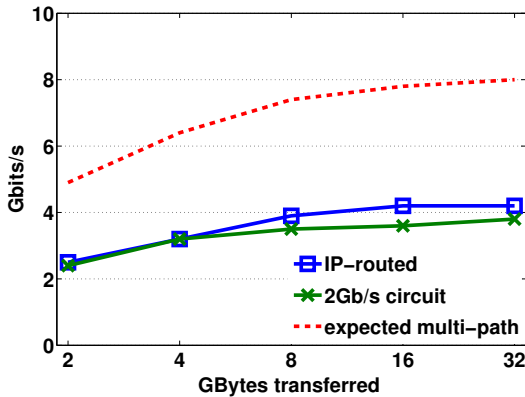


Fig. 3: Average performance for increasing transfer sizes between NERSC and ANL data transfer nodes. 10Gb/s shared IP and 10Gb/s circuit interfaces using 32 parallel streams.

This work has been motivated by an evaluation of the existing connectivity between end-sites with both shared IP and dedicated circuit capabilities. Results obtained from our preliminary GridFTP testing has demonstrated that making use of both the default, commodity IP routes in conjunction with dedicated circuits will provide a number of significant performance gains. Figure 2 shows the throughput of GridFTP transfers using 8 parallel streams between the 10G-connected DYNES installations at University of Michigan (UMich) and the California Institute of Technology (Caltech). Running two independent transfers simultaneously over both paths provides an indication of the expected multi-path throughput that could be achieved for a single transfer.

In Figure 3, we show the results of a similar test using 32 parallel streams between data transfer nodes located at NERSC and ANL. In each case, our reservable circuit capacity was limited to 2Gb/s because of capacity caps, although we note that due to bandwidth “scavenging” enabled in the Internet2 ION Service, we frequently see average rates above the defined bandwidth limit. This behavior is clearly indicated in our results.

While dedicated circuits provide a number of benefits in terms of deterministic characteristics and improved throughput for many data-intensive applications, a key drawback is

the often lengthy provisioning latencies. Building OSCARS circuits can frequently take on the order of 5-7 minutes as the number of domains involved in a path increases, requiring the application to idle while the path becomes available. One advantage of enabling simultaneous flows over both best-effort and circuit paths is the ability for a transfer to make progress while the circuit becomes active. A clear goal for our XSP dynamic network approach is to allow an SDN network to opportunistically redirect flows onto paths as they become available, providing a transparent mechanism that enables multi-pathing for data movement applications. Other advanced networks, e.g., NDDI/*OS³E*, promise to reduce path configuration time considerably through the use of OpenFlow, while still maintaining the ability to peer with OSCARS-based services such as ION.

A. Prior Work

Our earlier work with XSP [16] in dynamic network environments has demonstrated the ability for XSP to provide an application-driven interface to SDNs. At SC11, we used *xio-xsp* and XSPd deployed across remote domains to dynamically switch an active GridFTP transfer between shared (100G) and dedicated (10G) VLANs. The decision to redirect traffic was determined by a performance threshold integrated within the *xio-xsp* implementation, allowing the transfer to alternate between paths to achieve better transfer rates without input from the user. This work forms the basis of our multi-path experiments planned for SC12.

The Phoebus system has also been shown to support large scientific data transfers over reservable network paths using OSCARS [21]. We have demonstrated that on-demand network resources can provide significant performance gains for applications when compared to traditional direct connections. The ability of XSP, and hence Phoebus, to signal the OSCARS control plane allows PG deployments to immediately leverage these dynamic networks. In conjunction with SDN control at the edge, Phoebus will provide a WAN acceleration capability as part of our experiments.

Finally, in previous monitoring work [15], we showed results from the integration of Periscope for measuring end-to-end bottlenecks at gigabit transfer speeds. Our analysis demonstrated the ability to correctly aggregate and correlate metrics from GridFTP, hosts, and networks to accurately determine the cause of end-to-end performance bottlenecks. A graphical example of the bottleneck determination algorithm is shown in Figure 4. We intend to leverage this work to more intelligently influence the path selection used by our approaches described below.

B. Proposed Experiment

Our proposed experiment involves three distinct approaches to achieve multi-pathing with GridFTP. As our focus is on DYNES deployment, we will investigate the parallel use of both the best-effort, IP-routed path and the ability to dynamically provision and direct traffic over dedicated VLANs between available end-sites. In each scenario, we assume the presence of the following local area resources:

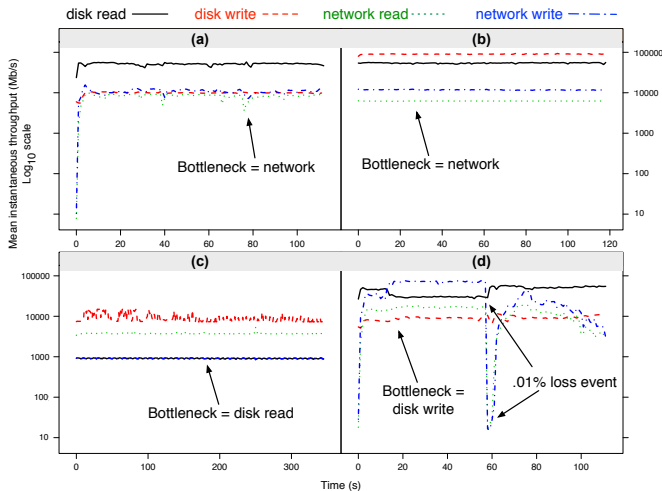


Fig. 4: Time series data representing GridFTP TCP transfer experiments: (a) 1 stream memory-to-disk with 100ms latency, (b) 1 stream memory-to-memory with no latency, (c) 1 stream disk-to-disk with no latency, (d) 4 streams memory-to-disk with 100ms latency and 1% loss added at 60 seconds.

- Dedicated data transfer nodes that run GridFTP servers in addition to our host monitoring agents.
- A system on which to run our XSPd controller with access to both the OSCARS Inter-Domain Controllers (IDCs) and OpenFlow network.
- The ability to poll switch counters or other real-time statistics from SCinet/SRS and the DYNES end-sites.

Over the wide-area, our experiment will make use of existing OSCARS circuit services to provision paths from DYNES end-sites to endpoints within SCinet. We also intend to leverage existing NDDI/OS³E capabilities for building paths over an OpenFlow backbone network. The following describes our three multi-path approaches in further detail.

1) *Enable GridFTP XIO multi-path for multi-homed hosts.* The most logical first step is to enable GridFTP to take advantage of multi-homed data transfer nodes. In the DYNES deployment, each node contains a 1Gb/s interface in addition to one or more 10Gb/s interfaces. OSCARS IDC deployments within the end-sites enable multi-domain circuits to extend directly to the data transfer nodes via dynamic VLAN creation across a local DYNES switch. One half of a typical DYNES scenario is depicted in Figure 5. To realize the the other half of our experiment setup, we will make use of the SRS OpenFlow network to extend the circuit termination point within SCinet to the data transfer node on the showroom floor. We will then be able to directly use an end-to-end “virtual circuit” in addition to the IP-routed path between each end-site.

As illustrated in the figure, provisioning the OSCARS circuit and installing flow entries into the SRS OpenFlow core is controlled by the XSPd instances running at both SCinet and within the remote DYNES end-site. Path creation is made possible by the *xio-xsp* driver enabled within the GridFTP servers, allowing each transfer to explicitly signal XSPd and control the network configuration for a given transfer operation.

The primary challenge in this scenario involves extending the XIO framework to take advantage of the multi-homed hosts. To instantiate parallel streams over each available interface, the XIO transport (e.g., TCP) driver will bind multiple data channels (one per interface) with the ability to establish connections over each path in turn. This can be done on the source without any changes to the GridFTP protocol. A client sends PASV to a destination server, the destination server starts listening on a port and sends the IP address and port to the client. The client then sends the IP address and port to the source server. The source server then binds to the same interface on which the control connection is received (or to a pre-configured interface) and establishes data channel connections with the destination server. We can modify this behavior and make the source server bind to different interfaces for each of the data channel connection.

The distribution of parallel streams across paths can be determined by the capacity of each link and the current utilization of each path based on network state provided by Periscope. On the destination side, we would have to use a striped server with multiple GridFTP data node instances, one bound to each active interface. This will enable multi-path transfer between the two end-sites. Another possibility is to use a striped server at both source and destination. In either case, we would have to use a new striping algorithm as the traditional one would always use all the pre configured data node instances. Here we would want to pick the interfaces dynamically based on the current state of network and other elements in the end-to-end path.

2) *Redirect to Phoebus and optimize transfers over the WAN.* Our second approach will explore the use of both the SRS and Phoebus to direct flows over alternate paths with the goal of accelerating these flows over the provisioned WAN segment. As shown in Figure 6, this experimental configuration uses a PG and the DYNES host as circuit endpoints, eliminating the extension of the circuit to the SCinet node. The PG software, known as *psd*, will be installed on the DYNES node to forward traffic arriving from the PG over the circuit to the publicly routed interface on the destination. As existing ION endpoints, we intend to use the PG deployment at Internet2 Points of Presence (POP) in our experiment, and we note that the PG in Salt Lake City is ideally situated given the SC12 locale. We will also explore the possibility of having a PG system available at SCinet as an additional adaptation point.

The key enabler of multi-path capability in this approach is the use of the SRS OpenFlow core to intercept some number of parallel GridFTP connections bound for the DYNES end-site and instead redirect them to the PG. Since the Phoebus architecture divides the end-to-end path into transport-specific segments, we will rely on the OpenFlow *Set-Field* action to rewrite the IP destination header fields for some n number of GridFTP streams to that of the PG. This effectively allows the GridFTP data source at SC12 to establish n connections to the PG, which are then forwarded to the destination hop over the WAN circuit, while the remaining connections are established over the best-effort path. The *xio-xsp* driver, being situated on the GridFTP XIO driver stack, is aware of the source information in the transport layer 5-tuple during data

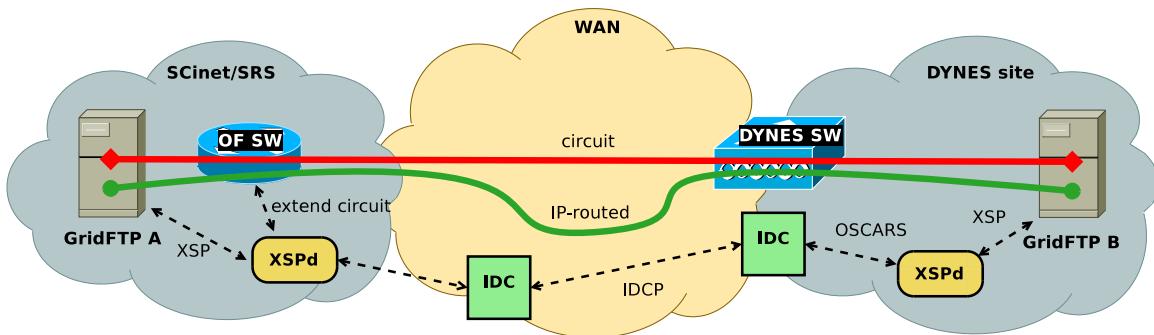


Fig. 5: *Approach 1*: GridFTP XIO takes advantage of multi-homed hosts and dedicated paths dynamically provisioned between each end-site.

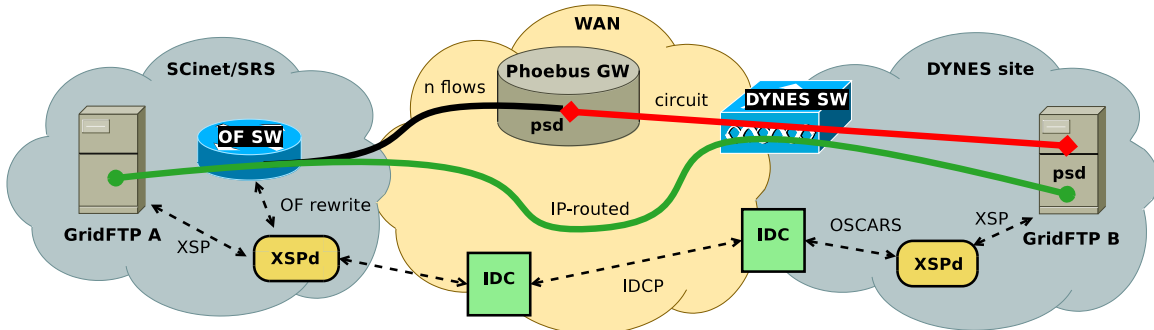


Fig. 6: *Approach 2*: XSPd configures OpenFlow switch to rewrite headers that match some or all GridFTP flows, sending them to a Phoebus Gateway that forwards and accelerates the transfer over the WAN circuit.

channel connection establishment. As such, *xio-xsp* is able to signal XSPd with the specific connection information (i.e. source IP and port number) that form the critical fields of the flow entries that govern which GridFTP flows are redirected to the PG.

By allowing the network to actively redirect flows, and by using Phoebus as a means to better utilize WAN circuits, we hope to improve upon the performance of purely end-to-end circuits between DYNES sites. Additionally, the level of performance achieved by the given OpenFlow switch implementation while rewriting headers is one of the metrics we will collect and analyze as a result of this experiment. We expect the major differentiating factor will be whether or not the *Set-Field* action, for the OpenFlow 12-tuple fields we are matching on, is implemented in hardware (i.e., on the ASIC versus forwarded over the management CPU).

3) *Dynamically redirect flows over VLANs with SDN*. Our last approach will investigate the use of the NDDI/*OS³E* OpenFlow backbone to provision alternate layer-2 paths. These paths are effectively dynamically configured VLANs across a network of OpenFlow switches. An NDDI configured “slice”, or flowspace, will be controlled by our XSPd instances from either end-site. As depicted in Figure 7, XSPd provides the network interface required to install flow entries into both the SRS and NDDI networks, enabling redirection of matching GridFTP streams onto the VLANs instantiated across the NDDI backbone.

Our effective evaluation of this scenario will depend on the availability of NDDI resources during SC12 in addition to any connectivity constraints in directing traffic to our remote end-sites. The ability of NDDI to peer with ION allows us to

extend these VLANs into the end-sites directly if necessary.

To summarize, our application-driven approach gives XSP-enabled transfer clients control over when and how to take advantage of network parallelism. In the simplest case, *xio-xsp* enables the automatic provisioning of circuits on behalf of GridFTP transfers that can make best use of the available end-to-end paths. Beyond observable performance improvements, what we expect to gain in each of our approaches is the ability to evaluate and tune how Periscope measurement feedback within our XSP-based services can determine when multi-pathing is beneficial. For example, if our analysis detects that a GridFTP transfer is disk I/O bound, then we have the potential to avoid wasteful provisioning of a dedicated circuit that would not help overall performance. Conversely, if we find that the network is the bottleneck, we have the option of finding alternate, or to allocate new, paths to support the transfer.

IV. RELATED WORK

There have been a number of proposed schemes for providing application-level multi-pathing [25]–[27]. While our approach shares common goals, our focus is on the use of active network signaling to provide alternative paths, and to actively redirect flows over dedicated wide-area paths.

In a similar vein, Resilient Overlay Networks (RONs) have been developed [7], [8] to provide and make better use of network path diversity. Overlays have also been applied specifically to improve GridFTP performance [14], [22].

Other clearly related work involves dynamic network resource allocation across both local and wide-area networking

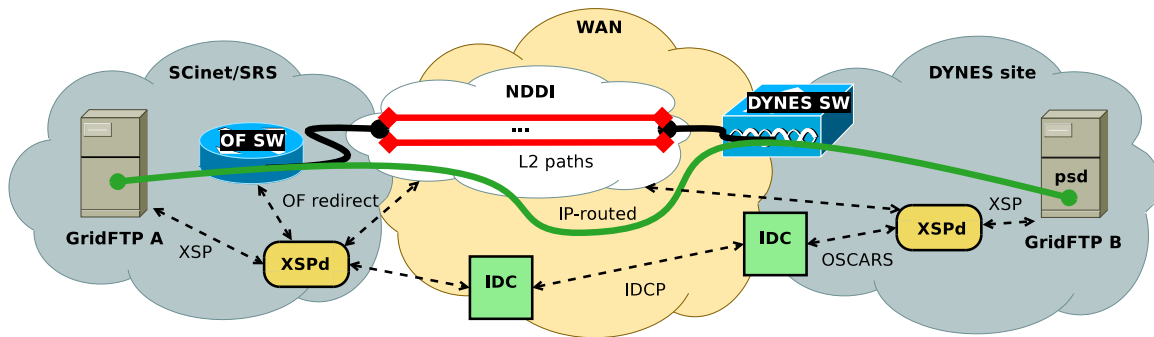


Fig. 7: Approach 3: The OpenFlow SRS core is configured to redirect flows along provisioned layer-2 paths within the NDDI OpenFlow backbone. XSPd also provides the interface through which the NDDI slice is controlled.

environments. Systems such as Terapaths [11] and LambdaStation [9] are two approaches designed to provide better access and use of dynamic network environments, which also directly use the OSCARS system for dynamic circuit provisioning.

V. CONCLUSION

This proposal has outlined three experimental approaches for achieving improved throughput via multi-pathing over advanced networks in support of Globus Transfer, and specifically, GridFTP. Our experiments are designed to leverage a number of existing technologies, including application-driven network configuration via XSP, GridFTP striping and parallelism, the Phoebus Gateway service, and Periscope monitoring and analysis capabilities. Each of our experiments is designed to make use of the SCinet SRS in conjunction with a combination of external WAN and SDN resources.

REFERENCES

- [1] Globus online — reliable file transfer. no it required. <https://www.globusonline.org>.
- [2] Network development and deployment initiative (nddi). <http://www.internet2.edu/network/ose/>.
- [3] Phedex cms data transfers. <https://cmsweb.cern.ch/phedex/>.
- [4] W. Allcock, J. Bester, J. Bresnahan, S. Meder, P. Plaszcak, and S. Tuecke. Gridftp: Protocol extensions to ftp for the grid. *Security*, (August):2010–2010, 2003.
- [5] W. Allcock, J. Bresnahan, R. Kettimuthu, and J. Link. The globus extensible input/output system (xio): A protocol independent i/o system for the grid. In *Protocol-Independent I/O System for the Grid. Joint Workshop on HighPerformance Grid Computing and High-Level Parallel Programming Models in conjunction with International Parallel and Distributed Processing Symposium*, 2005.
- [6] W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link. The globus striped gridftp framework and server. In *Proceedings of the ACM/IEEE SC 2005 Conference*, page 54, 2005.
- [7] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. Resilient overlay networks. Proc. 18th ACM SOSP, Banff, Canada, October 2001.
- [8] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. The case for resilient overlay networks. In *Proceedings of the 8th Annual Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, 2001.
- [9] A. Bobyshev, M. Crawford, P. DeMar, V. Grigaliunas, M. Grigoriev, A. Moibenko, D. Petravick, and R. Rechenmacher. Lambda station: On-demand flow based routing for data intensive grid applications over multitopology networks. In *Proceedings of the 3rd International Conference on Broadband Communications (IEEE)*, 2006.
- [10] A. N. S. for Information Processing Systems. Open systems interconnection – basic connection oriented session protocol specification. ANSI/ISO 8327-1987, 1992.
- [11] B. Gibbard, D. Katramatos, and D. Yu. Terapaths: A qos-enabled collaborative data sharing infrastructure for peta-scale computing research. In *Proceedings of the 3rd International Conference on Broadband Communications (IEEE)*, 2006.
- [12] A. Hanemann, J. Boote, E. Boyd, J. Durand, L. Kudarimoti, R. Lapacz, M. Swany, S. Trocha, and J. Zurawski. PerfSONAR: A service oriented architecture for multi-domain network monitoring. In *Proceedings of the Third International Conference on Service Oriented Computing (ICSOC 2005)*, ACM Sigsoft and Sigweb, pages 241–254, December 2005.
- [13] Internet2 ION. <http://www.internet2.edu/ion/>.
- [14] G. Khanna, U. Catalyurek, T. Kurc, R. Kettimuthu, P. Sadayappan, I. Foster, and J. Saltz. Using overlays for efficient data transfer over shared wide-area networks. In *SC '08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, pages 1–12, Piscataway, NJ, USA, 2008. IEEE Press.
- [15] E. Kissel, G. Fernandes, A. El-Hassany, M. Swany, D. Gunter, T. Samak, and J. Schopf. Scalable integrated performance analysis of multi-gigabit networks. In *5th Intl. Workshop on Distributed Autonomous Network Management Systems (DANMS)*, 2012.
- [16] E. Kissel, G. Fernandes, M. Jaffee, M. Swany, and M. Zhang. Driving software defined networks with XSP. In *Workshop on Software Defined Networks (SDN'12), Co-located with the IEEE International Conference on Communications (ICC)*, 2012.
- [17] E. Kissel and M. Swany. The extensible session protocol: A protocol for future internet architectures. Technical Report UDEL-2012/001, http://dams.cis.udel.edu/projects/phoebus/kissel_xsp.pdf.
- [18] E. Kissel, M. Swany, and A. Brown. Phoebus: A system for high throughput data movement. *J. Parallel Distrib. Comput.*, 71:266–279, February 2011.
- [19] N. McKeown, T. Anderson, H. Balakrishnan, G. M. Parulkar, L. L. Peterson, J. Rexford, S. Shenker, and J. S. Turner. Openflow: enabling innovation in campus networks. *Computer Communication Review*, 38(2):69–74, 2008.
- [20] ESnet On-demand Secure Circuits and Advance Reservation System (OSCARS). <http://www.es.net/oscars/>.
- [21] L. Ramakrishnan, C. Guok, K. Jackson, E. Kissel, D. M. Swany, and D. Agarwal. On-demand overlay networks for large scientific data transfers. *Cluster Computing and the Grid, IEEE International Symposium on*, 0:359–367, 2010.
- [22] P. Rizk, C. Kiddle, R. Simmonds, and B. Unger. Performance of a gridftp overlay network. *Future Gener. Comput. Syst.*, 24(5):442–451, 2008.
- [23] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288, 1984.
- [24] B. Tierney and D. Gunter. Netlogger: A toolkit for distributed system performance analysis. In *Proc. of the IEEE Mascots 2000 Conference (Mascots)*, 2000.
- [25] B. Wang, W. Wei, J. Kurose, D. Towsley, K. R. Pattipati, Z. Guo, and Z. Peng. Application-layer multipath data transfer via tcp: Schemes and performance tradeoffs. *Perform. Eval.*, 64(9-12):965–977, Oct. 2007.
- [26] J. Zhang, Y. Gui, C. Liu, and X. Li. To improve throughput via multipathing and parallel tcp on each path. In *Proceedings of the 2009 Fourth ChinaGrid Annual Conference, CHINAGRID '09*, pages 16–21, Washington, DC, USA, 2009. IEEE Computer Society.
- [27] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang. A transport layer approach for improving end-to-end performance and robustness using redundant paths. In *Proceedings of the annual conference on USENIX Annual Technical Conference, ATEC '04*, pages 8–8, Berkeley, CA, USA, 2004. USENIX Association.