# A Platform for Large-Scale Grid Data Service on Dynamic High-Performance Networks

**Tal Lavian[1], Doan Hoang[2], Joe Mambretti[3], Silvia Figueira[4], Sumit Naiksatam[4], Neena Kaushik[4], Monga Inder[1], Ramesh Durairaj[1], Doug Cutrell[1], Steve Merrill[1], Howard Cohen[1], Paul Daspit[1], Franco Travostino[1]**

[1] Nortel Networks Labs *{tlavian, imonga, radurai, pdaspit, smerrill, hcohen, dcutrell, travos}@nortelnetworks.com*
[2] University of Technology, Sydney *{dhoang@it.uts.edu.au}*[3] iCAIR, Northwestern University < *j-mambretti@northwestern.edu*>
[4] Santa Clara University *{sfigueira, snaiksatam, nrkaushik}@scu.edu*

***Keywords**:* Grid computing, OGSA, OASIS-WSRF, data-intensive applications, optical networking, dynamic provisioning, resource allocation, network resource scheduling.

***Abstract***— Data intensive Grid applications often deal with multiple terabytes and even petabytes of data. For them to be effectively deployed over distances, it is crucial that Grid infrastructures learn how to best exploit high-performance networks (such as agile optical networks). The network footprint of these Grid applications show pronounced peaks and valleys in utilization, prompting for a radical overhaul of traditional network provisioning styles such as peak-provisioning, point-and-click or operator-assisted provisioning. A Grid stack must become capable to dynamically orchestrate a complex set of variables related to application requirements, data services, and network provisioning services, all within a rapidly and continually changing environment. Presented here is a platform that addresses some of these issues. This service platform closely integrates a set of large-scale data services with those for dynamic bandwidth allocation, through a network resource middleware service, using an OGSA-compliant interface allowing direct access by external applications. Recently, this platform has been implemented as an experimental research prototype on a unique wide area optical networking testbed incorporating state-of-the-art photonic components. The paper, which presents initial results of research conducted on this prototype, indicates that these methods have the potential to address multiple major challenges related to data intensive applications. Given the complexities of this topic, especially where scheduling is required, only selected aspects of this platform are considered in this paper.

## 1. Introduction

The challenges related to data intensive Grid environments can be illustrated by an example involving an application's request to move a very large data file, which may exist as several copies, perhaps in several versions, at multiple networked locations across MAN/WAN, and within certain time windows. The file may be shared by multiple requesting processes and applications. This request must be filtered through policy agents, have its requirements validated, undertake a resource discovery process, have the appropriate resources linked to the request, and create processes that can be enacted by relevant local resource managers. To satisfy this application request, several issues have to be resolved, such as:

- Concurrent requests from multiple applications for data and/or storage.
- Storage space availability on the receiving side within the requested time window
- Critical data path not available, being used by other processes.
- Conflicting data transfer window with other requests.
- Limitation on currently available bandwidth.
- Inadequacy of resources and provisioning schemes to sustain bursty data movement with throughput of several Gb/s during selected time intervals.

The latter point proves to be an all important roadblock to success, when considering that:

- At multi-Gb/s rates, L3 network services (e.g., IP routing) prove sub-optimal in that they command high CapEx/OpEx costs, while their strength in many-to-many interactions is hardly a good match to the few-to-few patterns exhibited by Grid applications . Furthermore, they show many gaps in SLA provisioning across ISPs;
- The static provisioning of circuit-like dedicated resources would be highly inefficient, given the delta between peaks and valleys in network utilization;
- The dynamic provisioning of resources cannot possibly occur via point-and-click GUI interfaces, let alone operators (too fine a time scale; exceedingly error prone).

To satisfy various complex patterns of application demands, it is necessary to abstract and encapsulate the network resources into a set of Grid services that can provide scheduling, monitoring, and fair-shared usage within a service platform. Understanding application requirements and providing intelligence to respond to them are all standard issues that can be addressed within the context of Open Grid Service Architecture OGSA-compliant Grid middleware [1].

The paper describes an OGSA-compliant Grid-driven service platform that addresses these issues, with focus on orchestrating and integrating application requirements, large-scale data services, and agile provisioning services. The services components include mechanisms for application resource requests, resource discovery, and a specific set of defined data plus network provisioning services, with options for both on-demand and scheduled implementations. Each of these topics brings a special challenge. For example, scheduling requirements introduce difficult issues relating to variants in provisioning timeframes, from instantaneous to lengthy, among resources, e.g., data, network resources, and edge devices. Scheduling also can provide a means to predict future utilization among such resources.

The authors have adopted agile optical networks as the role-model of high-performance network that best matches the abovementioned traits of data intensive Grid applications. As such, the paper describes how optical network resources like optical fiber, lightpaths, and optical cross-connects switches are ultimately driven by the OGSA-compliant Grid infrastructure layers and network services. The authors have labeled this particular mapping "DWDM-RAM", to signify the coming together of optical resources and semantics as simple, accessible, dynamic, and popular as the ones of Random Access Memory. The resulting architecture, abstractions, and services are, however, directly applicable to other network resources outside of the optical realm.

In summary, the main contributions of the paper are:

1. Encapsulation of connection-oriented end-to-end network resources into a stateful Grid service, enabling on-demand, advanced reservation, and scheduled network services;
2. A schema wherein abstractions are progressively and rigorously redefined at each layer, so as to avoid gratuitous propagation of implementation-specific details adversely impacting portability. The resulting schema of abstractions has general applicability;
3. A stack whose layers are modeled after a "divide et impera" principle. For instance, network resources of the optical kind are first assembled and managed within a grid-unaware, general-purpose control plane specializing in routing, fault handling, impairment monitoring etc. Through this "separation of concerns", the layers are poised to scale to large topologies and large user communities, while leveraging legacy realities whenever applicable;
4. A prototype implementation of the architecture and its Network Resource Service (NRS) over an operational optical testbed, the OMNInet [2], and demonstrating its use via recurring file transfers;
5. Simulations through which it is possible to explore more complex scenarios in time/space. Through these simulations, overall utilization and blocking probability (the two inherent weaknesses to the connection-oriented network paradigm) can be assessed.

The rest of the paper is organized as follows. Section 2 presents relevant related research. Section 3 presents a proposed platform and its principal components. Section 4 describes the design of the Network Resource Service. Section 5 describes the prototype implementation. Section 6 describes the experimental testbed and discusses experimental results from the prototype implementation. Section 7 presents network scheduling and scheduling simulation results. Section 8 concludes the paper and discusses possible future work.

## 2. Related Research

The research described here is oriented specifically toward investigating the benefits of closely integrating Grid services, large-scale data flow capabilities, and agile optical networks [3].

This research is related to many other activities, including those related to the general Grid requirements for scheduling, e.g., allocating and reserving, multiple types of distributed resources, such as computational processing, data, instrumentation, and networks devices. Such resource management has always been the foundation of Grid research activities. The topics discussed here are also related to discussions on managing large scale data, including data-intensive collaboration [3], and data placement activities [4], and Data Grid projects [5, 6]. The importance of managing network resources for Grid computing has been central to these discussions initially introduced by Globus researchers, during the formative days of the I-WAY project.

The majority of initial research and development efforts related to Grid networking focused on integrated Grid services with enhanced L3-L4 protocols [7]. Many early efforts were focused on managing QoS traffic characteristics, for example, using layer-3 approaches to obtain QoS in packet switching networks [8, 9, 10, 11]. One such Globus project was GARA. Some efforts focused on integrating Grid services with enhanced L3-L4 protocols. [3, 12].

More recently, it has been recognized that some data-intensive Grid computing requires the handling of extremely large data sets, shared and distributed over multiple sites with such demanding high performance and service level guarantees that their requirements cannot be met by packet switched networks. This recognition has been presented in research workshop reports, in Grid standards development efforts (a draft describing issues related to optical network architecture for Grid services has been submitted to the GGF ) [13], and in special issues of scholarly publications.

Today, a number of research initiatives are focusing on integrating Grid services with emerging optical network capabilities [14, 15, 16, 17, 18]. Optical networks based on wavelength switching can be considered circuit switched and are able to provide high-bandwidth and layer-2 QoS easily, providing an additional resource option for data-intensive Grid computing. Furthermore, a number of optical testbeds have been established to support the research and development of these new architectures and platforms. They include OMNInet [2], the OptIPuter, a distributed Cyberinfrastructure being designed to support data-intensive scientific research and collaboration [19], I-WIRE, [20], DRAGON [21], experimental testbeds supported by

CA*net4 [22] which has introduced an innovative method for "User Controlled Lightpaths (UCLP) [23] and the Ultra Science Network [24].

Also, these new concepts are being demonstrated at national an international conferences, at iGRID2002, SC2003, and GGF. An architecture leveraging dynamic optical provisioning to bypass traditional networks was demonstrated in GlobusWorld 2004 [25].

Most significantly, these initiatives are directed at creating and implementing an architecture for networks based on dynamic wavelength utilization controlled at the network edge not within an centralized environment [22]. In part this concept of edge access represents a fundamental migration away from the legacy idea of managed network service within a heterogeneous centrally managed network to one that allows for highly distributed access to core network resources. These concepts are beginning to appear in national and international infrastructure, including in the TransLight [26], an innovative international network, the Global Lambda Integrated Facility (GLIF) [27], Ca*net4 [23], StarLight [28], NetherLight [29], UKLight [30], and others. It is expected that this architecture will also become part of large scale distributed computational infrastructure such as the TeraGrid [31].

## 3. Proposed Architectural Platform

Despite the progress represented by these projects, further efforts are needed before next generation optical networks can be used as a "first class" resource, enabling them to be used widely to support Grid applications. As noted, this is a complex issue, and many research projects exist that are directed at addressing one or more aspects of the problem. This paper describes one of the few research initiatives that address both the data services and the dynamic transport issues. Specifically, based on a set of experiments and research results, this paper proposes an architectural platform that was designed to assist in progressing toward the goal of defining a data intensive service based on dynamic optical networking.

This platform that separates two sets of concerns: one deals with the application data and user requests and the other deals with the underlying resources. As shown in Figure 1, the proposed architecture identifies two distinct planes over the underlying dynamic optical network: 1) the Data Grid Plane that speaks for the diverse requirements of a data-intensive application by providing generic data-intensive interfaces and services and 2) the Network Grid Plane that marshals the raw bandwidth of the underlying optical network, exposes it as a set of network services, within the OGSA framework, and matches the complex requirements specified by the Data Grid Plane.

In its most general form, the middleware block diagram is shown in Figure 2. Two service layers, application middleware layer and the resource middleware layer, lie between an application and the underlying optical network. The application middleware layer presents a service interface to users/applications and understands the requirements of the application. This layer also shields the application from all complexities of sharing and managing the required resources. At this layer, low level services offered by Grid resources in the resource middleware layer are orchestrated into high level services such as work flow, service discovery, replication, data transfer, etc.

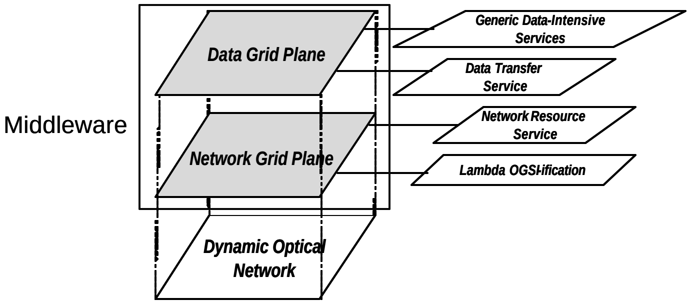

**Figure 1** – A plane view of the middleware architecture

The resource middleware layer provides services that satisfy the resource requirements of the application, as specified or interpreted by the application middleware layer. This layer presents a service interface thus abstracting the details concerning specific underlying resources and switching technologies (e.g., lambdas from wavelength switching, optical bursts from optical burst switching, etc.) to the layer above. This layer contains capabilities that initiate and control sharing of the underlying resources as well as service components for managing Grid resources such as network, processing, storage, and data handlers.The underlying network and its associated protocol provide the connectivity and fabric for the application.

This paper addresses concerns around the role of network services in a scheduled transport of data between application end points. Specifically proposed is an architecture that addresses the complex integrated issues mainly concerning scheduling network resources and data transfers. For this reason, only selected components of the general architecture are discussed and prototyped in our DWDM-RAM architecture.

### *3.1 Application Middleware Layer*

At the application middleware layer, the Data Transfer Service (DTS) presents an interface between the system and an application. It receives high-level client requests, policy-and-access filtered, to transfer named blocks of data with specific advance scheduling constraints. It employs an intelligent strategy to schedule an acceptable action plan that balances user demands and resource availabilities. The action plan involves advance co-reservation of network and storage resources. This middleware layer shields the application from lower level details by translating application-level requests to its own tasks of coordinating and controlling the sharing of a collective set of resources.

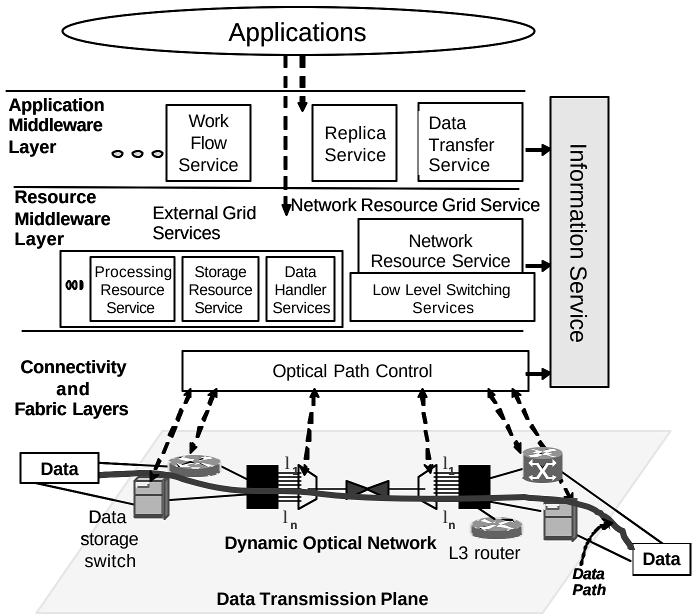

**Figure 2** – A generic middleware system architecture

### *3.2 Network Resource Middleware Layer*

The network resource middleware layer consists of three services: the Data Handler Service (DHS), the Network Resource Service (NRS) and the Dynamic Lambda Grid Service (DLGS). Services of this layer initiate and control sharing of resources. The DHS deals with the mechanism for sending and receiving data and effectuates the actual data transfer when needed by the DTS. A central piece of this layer, the Network Resource Service (NRS), makes use of the Dynamic Lambda Grid Service in encapsulating the underlying optical network resources into an accessible, schedulable Grid service. The NRS receives requests from the DTS, as well as requests from other services such as external Grid services (both scheduled and on-demand). It maintains a job queue and allocates proper network resources according to its schedule. To allow for extensibility and reuse, the Network Resource Service can be decomposed into two closely coupled services: a Basic Network Resource Service and a Network Resource Scheduler. The Basic Network Resource Service handles multiple low level services offered by different types of underlying networks and switching technologies and presents an interface to the Data Transfer Service for making network service requests. The Network Resource Scheduler is responsible for implementing an effective schedule that facilitates network resources sharing among multiple applications. The Network Resource Scheduler can be deployed independently of the Basic Network Resource Service. This provides the NRS the flexibility to deal with other scheduling schemes as well as other types of dynamic networks. This paper only considers a simple scheduling scheme and a wavelength (lambda) switching service (DLGS) over a dynamic optical network. The Dynamic Lambda Grid Service receives resource requirement requests from the NRS and matches those requests with the actual resources, such as path designations. It has complete knowledge of network topology and network resource state information because it receives this information from lower level processes. The Dynamic Lambda Grid Service can establish, control, and deallocate complete paths across both optical and electronic domains. The detailed NRS architecture and design are discussed in the next section.

These layers may also communicate with an information service or services, in order to advertise their resources or capabilities.

DWDM-RAM [18], our presented architecture, matches well with the Layered Grid Architecture [1] in Figure 3. The DTS sits at the Collective Layer, coordinating multiple resources; the NRS/DLGS is at the Layered Grid Architecture's Resource Layer, controlling the sharing of single network resources; and the Optical Path Control is for connectivity provisioning. The architecture is sufficiently flexible such that it can be implemented as layers or modules via an object approach.

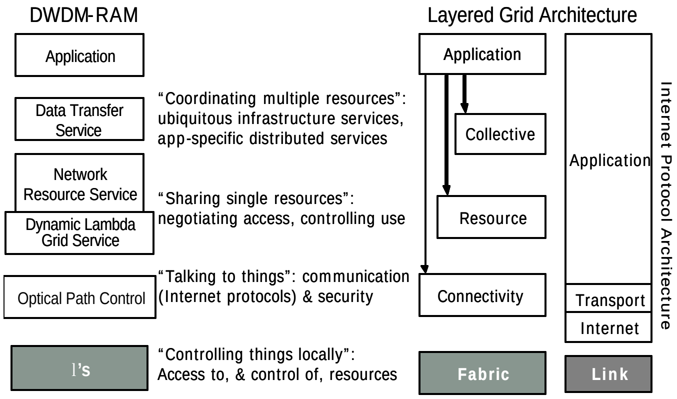
DWDM-RAM
Layered Grid Architecture
Application
Data Transfer Service
Network Resource Service
Dynamic Lambda Grid Service
Optical Path Control
l's
"Coordinating multiple resources": ubiquitous infrastructure services, app-specific distributed services
"Sharing single resources": negotiating access, controlling use
"Talking to things": communication (Internet protocols) & security
"Controlling things locally": Access to, & control of, resources
Application
Collective
Resource
Connectivity
Fabric
Application
Transport
Internet
Link
Internet Protocol Architecture

**Figure 3** – DWDM-RAM versus Layered Grid Architecture (adapted from [3])

## 4. Design of the Network Resource Service

There are three primary goals for the design of the Network Resource Service: 1) provide a high-level interface for applications and services to allocate bandwidth; 2) provide transparent interfaces to a variety of underlying network control mechanisms; and 3) provide reservation and scheduling functionality.

A usable application-level interface is essential to making dynamically provisioned high-bandwidth optical network resources available to a wide variety of applications, while hiding the complexities of the underlying network and network control mechanisms. Unlike packet switched networks, a dynamically provisioned wavelength is dedicated to only one user at a time. Thus efficient utilization of dynamic wavelength provisioning requires scheduling, both to optimize the utilization of the underlying resources, and to provide applications with predictable and reliable bandwidth.

### *4.1 Scheduling Services*

Although the Network Resource Service supports both on-demand and scheduled allocations, scheduling is perhaps its most significant feature, and an integral part of the design. The scheduling of lightpaths is key to guaranteeing that the network will be available when needed.

Grid computing involves the simultaneous usage of different resources connected by network links. This characteristic of most grid computing mandates that all the resources involved in a computation be available at pre-determined times. For example, if a surgery is to be performed remotely by a surgeon located in another location, it is really crucial that the network be available at full speed at the time for which the surgery was scheduled. To enable that, it is important to have a scheduler that guarantees the availability of the network. If not, there is a chance that the network may not be available at the right time. On the other hand, the interfaces for requesting bandwidth should be rich enough to allow clients to express their own flexibility via under-constrained requests. This allows for optimal scheduling, and for automatic rescheduling as well. The network scheduling authority considers the flexibility in the requests, the flexibility inherent in any conflicting current reservations, and other factors such as job priorities or predictive load balancing. It provides guarantees or reservations for channel availability between specific endpoints, with certain constraints. These reservations are controlled and accessed through tickets which can be shared between applications. The reservations may be periodic, and may be many days in advance. They may be policed by various resource reclamation policies, such as periodic application resiliency requirements. When rescheduled within the bounds of the under-constrained request to meet new requests, these changes are reported via a client query or notification mechanism.

Middleware services, such as data transfer services, need to coordinate their activities very precisely with network allocations. Therefore, the underlying scheduler used as the network authority can also be used to drive schedules exposed by other services. These services may add their own request language and do initial pre-processing of the request before translation to the network layer. For example, a data transfer service may receive requests which specify data in some abstract fashion. The data transfer service may consult a replica location service in order to determine the actual endpoints, and it computes the length of the requested network reservation based on the requested file size.

For network scheduling purposes, each lightpath on each network segment is viewed as a sharable resource which can be dedicated for only one use at a time. The scheduler can also link these lightpath segments into scheduled end-to-end lightpaths. It is generally the bandwidth provided by these end-to-end lightpaths that is of interest to applications, so the interface hides the underlying segment-level scheduling from callers that don't need it.

The scheduling of network paths requires coordination of multiple resources with heterogeneous constraints. For example, different segments may lie in different domains of administrative control. Some segments may have constraints such as limited windows of availability, or high costs. When multiple paths are possible, the choice of segments allows a further area of optimization. All of this may be reconsidered as the system attempts rescheduling to satisfy new requests. The extension of the scheduling problem to higher level services such as data transfer introduces additional constraints for optimization.
Some application services which use the network scheduler may allow additional avenues for optimization. To take advantage of this, the network scheduler allows clients to request that they be included in a voluntary rescheduling protocol. Via this protocol, a client is requested to consider rescheduling a previously granted reservation. For example, a data transfer service may be able to source data from a different location in order to free up a network segment which is needed by some other service during the currently reserved time.

## 5. Prototype Implementation

We have implemented prototypes of the Network Resource Service (NRS), the Data Transfer Service (DTS), and the Data Handler Service (DHS). The primary goal for this implementation was to test the architectural concepts described above on a real metro-area optical network. In particular, we wanted to 1) implement a suitable application-level interface to the Network Resource Service for allocating bandwidth to applications; 2) develop a resource scheduling infrastructure suitable for optical networks and other resources; 3) provide for interoperability with Grid applications and services using OGSA/OGSI standards; and 4) collect experimental data and analyze Grid data-intensive service over a metro area network.

### *5.1 Network Resource Service Interface and Functionality*

The implementation of the NRS presents client applications and services with an interface for advanced reservation of lightpaths with guaranteed bandwidth. Lower-level details related to the optical network are hidden from the application.

To request a lightpath reservation, a client application specifies 1) the two hosts it wants to connect, 2) the duration of the connection, and 3) the time window in which the connection can occur, specified by the starting and ending time of the window.

The NRS returns a “ticket” describing the resulting reservation, if it is possible to make one meeting the given requirements. This ticket includes the actual assigned start and end times, as well as the other parameters of the request. The ticket can be used in subsequent calls to change, cancel, or obtain status on the reservation. The NRS will allocate the indicated lightpath at the agreed-upon time as long as the reservation has not been canceled or changed since it was made.

Sample pseudo code for a client calling the NRS is shown in the following figure:

```
// Bind to an NRS service:
NRS = lookupNRS(address);
//Request cost function evaluation
request = {pathEndpointOneAddress,
           pathEndpointTwoAddress,
           duration,
           startAfterDate,
           endBeforeDate};
ticket = NRS.requestReservation(request);
// Inspect the ticket to determine success, and to find
the currently scheduled time:
ticket.display();
// The ticket may now be persisted and used
from another location
NRS.updateTicket(ticket);
// Inspect the ticket to see if the reservation's scheduled time
has changed, or verify that the job completed, with any
relevant status information:
ticket.display();
```

**Figure 4** – NRS client pseudo-code

### *5.2 Data Transfer Service*

The Data Transfer Service (DTS) is a middleware service built on top of the NRS. It provides an interface for applications to request the transfer of named data sets from one location on the Grid to another. In the current implementation the source host and pathname are explicitly required. In the future, with the use of a replica location service, only an abstract source data set name will be required and DTS will choose an appropriate physical source location. Data transfer requests can also specify scheduling parameters such as window start and end times, and transfer duration.

DTS does some processing before calling on the NRS scheduling facility. This includes finding the appropriate data source; verifying its existence, size, and accessibility; coordinating with the destination system storage scheduler, and the like. The client's request, possibly modified by these additionally discovered constraints, is passed to the NRS scheduler, and the received job ticket passed back to the client. A cron-like entry is made for DTS to wake up at the scheduled network allocation and data transfer time.

At the scheduled transfer time, the DTS sends a message to a Data Handler Service (DHS) running on the destination host, which then opens a connection to the source host and transfers the data. Status is passed back to the DTS and is available to client queries and for registered client callbacks.

### *5.3 Prototype Grid Integration*

This platform anticipates access to resources within the context of the Open Grid Services Architecture (OGSA). This platform will be interoperable with Grid applications that can utilize WS-Agreements to negotiate services to satisfy the requirements and network characteristics of high throughput file transfer.

Therefore, the first implementation was based on fully-OGSI compliant Grid Service interfaces for use by Grid applications. This interface is being migrated to WSRF.

These interfaces were implemented using Version 3 of the Globus Toolkit [32], using SOAP and the JAX-RPC API. Additionally, the interfaces are wrapped in Java classes that effectively hide the OGSI middleware from the applications that call them.

These interfaces are intended to provide the applications that call them with the basic functionality they need, but without requiring any knowledge of the underlying networks or network control and management protocols.

## 6. Testbed and Experimental Results

### *6.1 Optical Dynamic Intelligent Network Services (ODIN)*

Optical Dynamic Intelligent Network Services (ODIN) is an architecture that provides for receiving communication service requests from higher level processes and translating those requests into network resources, primarily dynamically provisioned paths lightpaths and extensions of those lightpaths to edge devices through VLANs [14, 15, 16, 17] ODIN has been designed for extremely high performance, long term data-intensive flows requiring flexible and fine grained control. ODIN also receives network state information through various channels from the optical network core. Currently, it is implemented as server software that is comprised of several components, including APIs. One initiates processes by accepting requests from clients for network resources. The client requests activity that implies a request for a path or paths to resources. Given request attributes and complete knowledge of available network resources, ODIN designates appropriate paths. ODIN also creates the mechanisms required to route the data traffic over the defined optimal path (virtual network), and transmits signals that notify the client and the target resource to adjust to match the configured virtual network. An implementation of ODIN has successfully been used for large scale experiments with science data on the OMNInet testbed (described below) [14, 15, 16, 17]. Although this architecture was initially developed as a central control mechanism for intra-domain processing, it was designed such that it can be also be used to support distributed management and control functions and inter-domain processes. Like UCLP architecture, this approach is distinct from that being taken by ITU standard development efforts which are oriented toward centralized management and control. [33].

### *6.2 OMNInet testbed*

The OMNInet project is a multi-organizational partnership, which was established to build an advanced metro area photonic network testbed [17]. OMNInet is a wide area testbed consisting of four photonic nodes at widely separate locations in the Chicago metro area. These nodes are interconnected as a partial mesh with lightpaths provisioned with DWDM on dedicated fiber. Each node includes a MEMS-based (Micro-Electro-Mechanical Systems) Wave Division Multiplexed (WDM) photonic switch, Optical Fiber Amplifiers (OFAs), optical transponders/receivers (OTRs), and high-performance L2/L3 router/switches with 10Gb/s line-side Ethernet interfaces. The core photonic nodes are not commercial products but unique experimental research implementations, integrating state of the art components. The photonic switches are supported by high L2/L3 switches, which are provisioned with 10/100/1000 Ethernet user ports.

A separate OMNInet control plane is provisioned out-of-band using completely separate fiber, Such control planes could also reside on a supervisory lightpath. This control plane enables User-to-Network Interface (UNI) control signaling via a UNI interface to the optical transport network and bi-directional signaling to the connection control plane. 10GigE trunk interface using 15xx nm DWDM wavelengths have been implemented, with a specialized set of protocols that allows for enhanced optical network intelligence, including a lightpath signaling protocol, a lightpath routing protocol, and an optical link management protocol. To provide for reliability and optimal L1 performance, OMNInet is provisioned with sophisticated pre-fault detection mechanisms, which monitor network conditions and adjust resources in response to specific detected characteristics.

### *6.3 Experimental Results*

The primary objective of these experiments is to demonstrate that the underlying connection-oriented end-to-end network resources can be encapsulated within a network resource service that offers both on-demand and scheduled network services to Grid applications. Here, the DMDW-RAM architecture was deployed to demonstrate data-intensive file transfer and memory transfer services over a high-speed speed optical networking testbed to requesting applications.

#### *6.3.1 Scheduling*

An early prototype scheduling service was demonstrated at the SuperComputing 2003 conference in Phoenix, Arizona, 17–21 November, 2003.Several applications may request data transfers that involve the use of the same network segment for durations which may overlap. Figure 5 illustrates how the use of under constrained requests and rescheduling works by considering requests for a single segment. In Figure 5a, initially, a request has been made for a 70 minute block (A), sometime between 4:00 pm and 8:10 pm. Since there were no previous requests, it was granted for the first 70 minutes in this time window.

In Figure 5b, a second request (B) was made. It was in the 4:15 to 7:00 pm range and for 105 minutes. In order to meet B’s request the Scheduler placed it at the beginning of its time window and moved A, the first request, to immediately follow the second request’s completion. The first request, A, still satisfied its under-constrained window.

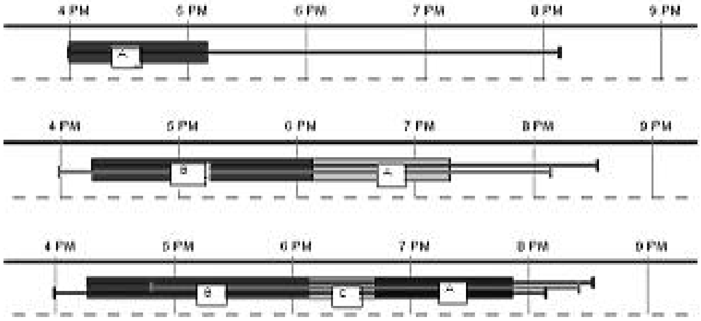

Figure 5a, 5b, 5c – Behavior of the scheduling algorithm as three successive requests are made to use one segment.

In Figure 5c, a third request (C) was made, between 4:45 and 7:00 pm for 30 minutes. Again, A was rescheduled to accommodate C, and B was left at its original allocated time.

The results demonstrated that with the network resource service, application requests can be scheduled and rescheduled to make efficient use of network resources.

#### *6.3.2 End-to-End File Transfer (Nonoptimized)*

This section demonstrates that by deploying our service platform, connection-oriented end-to-end data-intensive file transfer operations can be set up and executed on-demand easily and effectively.

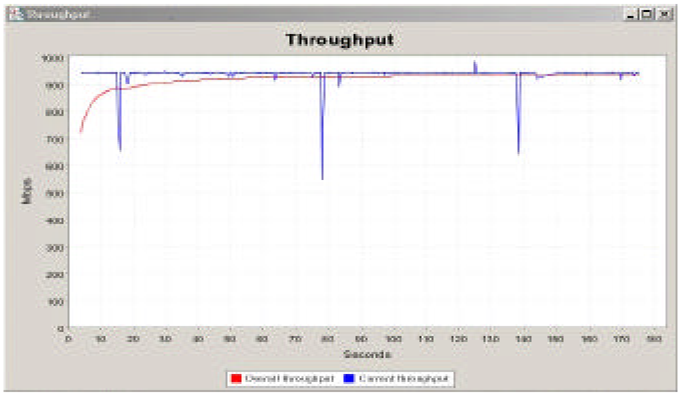

**Figure 6** – The throughput recorded for a 20 GB file transfer. The effective file transfer rate measured over OMNInet was 920 Mbps using *ftp*.

The preliminary file transfer results presented in Figure 6 were obtained from 20GB file transfer experiments performed on the OMNInet testbed and were demonstrated during GGF-9, October 2003, in Chicago, and SuperComputing, November, 2003 in Phoenix. The demonstrations were run on dual PIII 997 MHz machines, running Red Hat Linux 7.3 (Kernel: 2.4.18-3smp) and using 1 GigE NIC cards.

Table 1 shows the breakdown of times measured for many of the steps in the process. The bulk of the time was spent in the actual file transfer, but there was a significant cost of about 25 seconds to configure and 11 seconds later on tear down the lightpaths. It is important to note that only a minuscule fraction (approximately 20 milliseconds) of this delay is actually due to the setting up of the path within the optical switch. The major portion of the delay is introduced by switches at the edge of the optical network, which switch the data to and from the end hosts. This is an artifact of the current network setup. Another contributor to this set up delay is disk access when reading configuration files, and when writing updated switching tables to the disk.

Current path tear down is 11 seconds, which is due to the centralized path update from all optical nodes. With fine-tuning we expect to reduce substantially the network configuration and the path release overheads.

| Event | Seconds |
|---|---|
| Start : File transfer request arrives | 0.0 |
| Path Allocation request | 0.5 |
| ODIN server processing | 3.6 |
| Path ID returned | 0.5 |
| Network reconfiguration | 25 |
| FTP setup time | 0.14 |
| Data transfer (20GB file) | 174 |
| Path deallocation request | 0.3 |
| Path tear down | 11 |

**Table 1.** Breakdown of end-to-end file transfer time

It should be emphasized that the focus of paper is not on exploring various schemes for maximizing the data transfer throughput, rather it focuses on encapsulating the underlying network resources and offering network scheduled services to an application. For this reason we have only used the standard TCP based networking protocol stack readily available in the Linux kernel. The OMNInet testbed network guarantees high bandwidth channels since the network setup allocates a lambda of 10Gbs to each data flow. In spite of the network supporting this very high bandwidth, we observed that the end-to-end data flow between hosts does not fully utilize this bandwidth. As the OMNInet is inherently reliable with extremely low bit error rates and fairness is not an issue (dedicated lambda per flow), it is an overkill using the reliable TCP as the transport protocol for such data-intensive applications. Rather a performance tuned transport protocol like FAST [12] or SABUL/UDT [7] tailored to benefit form circuit switching path with no L3 routing, would be more effective in achieving high throughput over the 10Gbps pipe.

## 7. Network Scheduling

A Grid Scheduled Network Service is just like any other Grid service; it has to expose its service through a Grid interface. For example, an interface for requesting lightpath connectivity should be rich enough to allow applications to express their own flexibility via under-constrained (or loose-constrained) requests. This allows for optimal scheduling, and for automatic rescheduling if necessary. The scheduling network service considers the flexibility in the requests, the flexibility inherent in any conflicting current reservations, and other factors such as job priorities or predictive load balancing. It provides guarantees or advance reservations for channel availability between specific endpoints, with certain constraints. The reservations may be periodic, and may be many days in advance. They may be policed by various resource reclamation policies, such as periodic application resiliency requirements. When rescheduled within the bounds of the under-constrained request to meet new requests, these changes are reported via a middleware query or notification mechanism.

### *7.1 Simulation Results*

To illustrate the importance of scheduling the network, the graph in Figure 7 shows the probability that a request may not be satisfied, i.e., the network blocking probability, obtained both by simulation and by the Erlang B model. The simulation was driven by a trace generated by our Flexible Optical Network Traffic Simulator, FONTS [34]. In the simulation, each experiment consists of requests generated for one network segment, which contains 4 wavelengths, during one week. Each request is for one-hour slot, and the experiments presented in the graph are described in Table 2. The generation of requests follows two independent Poisson distributions: one for the arrival of requests and another for the reservation time.

| Experiment | Inter Arrival Time | Number of Wavelengths |
|---|---|---|
| 1 - High traffic with constant wavelength requests | Requests: 33.3min<br>Reservation: 10min | Constant: 1 |
| 2 - High traffic with uniform wavelength requests | Requests: 33.3min<br>Reservation: 10min | Uniform: [1-4] |
| 3 - Medium traffic with constant wavelength requests | Requests: 33.3min<br>Reservation: 25min | Constant: 1 |
| 4 - Medium traffic with uniform wavelength requests | Requests: 33.3min<br>Reservation: 25min | Uniform: [1-4] |
| 5 - Low traffic with constant wavelength requests | Requests: 33.3min<br>Reservation: 50min | Constant: 1 |
| 6 - Low traffic with uniform wavelength requests | Requests: 33.3min<br>Reservation: 50min | Uniform: [1-4] |

**Table 2:** Experiments presented in Figures 7 and 8 .

Note that we define traffic to be the frequency at which the requests happen. As shown in Figure 7, the blocking probability depends on the frequency of requests and can be as high as ~76% when the traffic is high, which indicates that reserving the network in advance is key to guaranteeing its availability.

Our scheduler supports both constrained and under-constrained requests. Under-constrained requests are useful for those applications that may have some flexibility. Using the surgery scenario of section 4.1, assume the surgeon is available during 5 hours in the morning for a one-hour surgery. In this case, the network request should use an under-constrained window. However, after the reservation is made for a specific hour, within the 5-hour window, that should become a hard commitment.

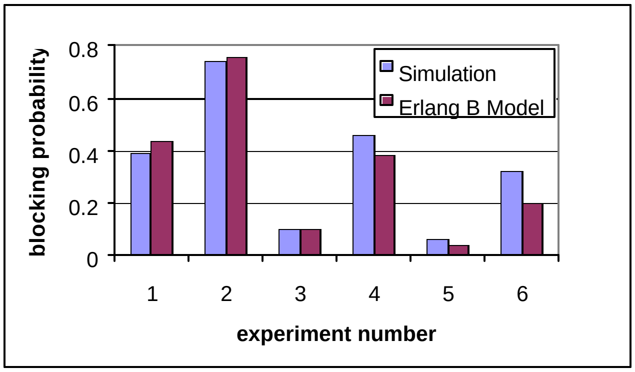

**Figure 7:** Blocking probability for different types of sets of requests.

Under-constrained requests give the scheduler more flexibility, allowing it to accommodate more requests, which results in a lower blocking probability and, consequently, in a better utilization of the network. The graph in Figure 8 illustrates the effect of under-constrained requests. It shows the blocking probability for the same experiments described in Table 2, when a window of 5 hours is used. The first column shows the blocking probability when no under-constrained requests are used, i.e., 0% of the requests had a window larger than the time slot requested. The second column shows the blocking probability when 50% of the requests had a 4-hour window following the 1-hour time slot requested. The third column shows the blocking probability when 100% of the requests had a 4-hour window following the 1-hour time slot requested. According to the graph, using under-constrained requests provides a gain in all the experiments performed. However, the gain is higher when the traffic, i.e., the frequency of requests, is medium or low. This shows that offering the under-constrained option and encouraging its usage (for example, by charging less from more flexible users) will be beneficial. Note that, with medium and low traffic, when the number of wavelengths is constant, the blocking probability decreases linearly with the number of under-constrained requests.

The graph in Figure 8 also shows, in the fourth column, the blocking probability's lower-bound, obtained by the Erlang B model when 100% of the requests had a 168-hour window, i.e., each request could be allocated in any slot during the week. This shows maximum flexibility and, consequently, minimum blocking probability. Note that, in five of the experiments, the blocking probability in the third column (where 100% of the requests are under-constrained) is close to the lower-bound value, which indicates that, in this case, a 5-hour window brings down the blocking probability considerably. Note also that, for low or medium traffic, in experiments 3, 5, and 6, the blocking probability is zero or close to zero when 100% of the requests are under-constrained.

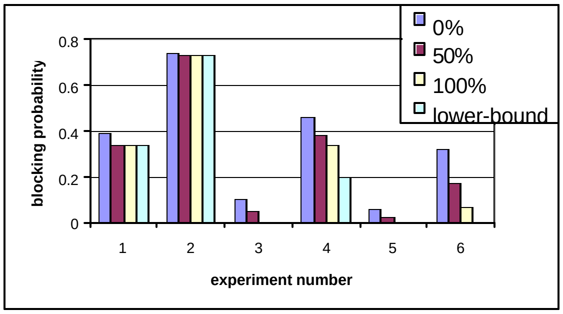

**Figure 8:** Blocking probability when under-constrained requests are used.

## 8 Conclusion

The paper describes a platform capable of forging close cooperation between data intensive Grid applications and network resources. Several layers mediate between the former and latter, while abstracting out low-level resources, and yielding opaque "tickets" to upper layers. Early standardization work around OGSA has shaped some of the layers' interfaces.

The authors have reduced to practice the platform over a best-of-breed agile optical network (hence the DWDM-RAM label for this particular mapping). This exercise validates the fit with data intensive Grid applications. Furthermore, it gives a quantitative read of "bit-blasting" behaviors as well as the all-important "finesse" aspects of a highly dynamic system (e.g., a lightpath's setup and tear-down overhead). Some simulation results around schedulability and blocking probability complement the empirical measurements, and reach out to scenarios of far greater complexity.

Throughout its layers, the platform is well suited to "top-half" optimizations catering to applications, and "bottom-half" evolutions focusing on the network element (e.g., to exploit more diverse networks).

## Acknowledgments

This research was made possible with support from DARPA, award number F30602-98-2-0194. ODIN research was made possible with support from the National Science Foundation: award number ANI-0123399.

The authors would like to acknowledge the contributions of Fei Yeh and Aaron Johnson of Northwestern University installing, configuring, testing, and measurements over the OMNInet.

## References

[1] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," *Open Grid Service Infrastructure WG*, Global Grid Forum, June 2002. (Available at http://www.globus.org/research/papers/ogsa.pdf)

[2] http://www.icair.org/omninet

[3] I. Foster, A. Roy, V. Sander, "A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation", *8th International Workshop on Quality of Service (IWQOS 2000), pp. 181-188, June 2000*.

[4] T. Kosar and M. Livny, "Stork: Making Data Placement a First Class Citizen in the Grid," *Proceedings of 24th IEEE Int. Conference on Distributed Computing Systems (ICDCS2004)*, March 2004.

[5] I. Foster, J. Vockler, M. Wilde, and Y. Zhao, "The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration," *Proceedings of the First CIDR - Biennial Conference on Innovative Data Systems Research*, January 2003.

[6] http://web.datagrid.cnr.it

[7] Y. Gu, R. Grossman, "SABUL (Simple Available Bandwidth Utilization Library)/UDT (UDP-based Data Transfer Protocol)," http://sourceforge.net/projects/dataspace,

[8] I. Foster, A. Roy, and V. Sander, "A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation," *8th International Workshop on Quality of Service*, 2000.

[9] G. Hoo, W. Johnston, I. Foster, and A. Roy, "QoS as Middleware: Bandwidth Reservation System Design," *Proceedings of the 8th IEEE Symposium on High Performance Distributed Computing*, pp. 345-346, 1999.

[10] V. Sander, W. A. Adamson, I. Foster, and A. Roy. "End-to-End Provision of Policy Information for Network QoS," *Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC-10)*, August 2001.

[11] V. Sander, I. Foster, A. Roy, and L. Winkler, "A Differentiated Services Implementation for High-Performance TCP Flows," *Elsevier Computer Networks*, vol. 34, pp. 915-929, 2000.

[12] Fast Active queue management Scalable Tcp. See http://netlab.caltech.edu/FAST/

[13] D. Simeonidou (Ed), "Optical Network Infrastructure for Grid," Draft submitted at *GGF 9*, Chicago, Oct 5 – 8, 2003. (http://forge.gridforum.org/projects/ghpn-rg/).

[14] J. Mambretti, J. Weinberger, J. Chen, E. Bacon, F. Yeh, D. Lillethun, B. Grossman, Y. Gu, M. Mazzuco, "The Photonic TeraStream: Enabling Next Generation Applications Through Intelligent Optical Networking at iGrid 2002," *Journal of Future Computer Systems, Elsevier Press*, August 2003, pp.897-908.

[15] R. Grossman, Y. Gu, D. Hamelberg, D. Hanley, X. Hong, J. Levera, M. Mazzucco, D. Lillethun, J. Mambrett, J. Weinberger, "Experimental Studies Using Photonic Data Services at iGrid 2002," *Journal of Future Computer Systems, Elsevier Press*, August 2003, pp.945-956.

[16] R. Grossman, Y. Gu, D. Hanley, X. Hong, J. Levera, M. Mazzucco, D. Lillethun, J. Mambretti, J. Weinberger, "Photonic Data Services: Integrating Path, Network and Data Services to Support Next Generation Data Mining Applications, *NGDM'02, Proceedings*, November 2002.

17] T. DeFanti, Brown, M., Leigh, J., Yu, O., He, E., Mambretti, J., Lillethun, D., and Weinberger, J. "Optical switching middleware for the OptIPuter," in a special issue on photonic IP network technologies for next-generation broadband access. *IEICE Transact. Commun*. E86-B, 8 (Aug. 2003), 2263--2272.

[18] S. Figueira, S. Naiksatam, H. Cohen, D. Cutrell, D. Gutierrez, D. B. Hoang, T. Lavian, J. Mambretti, S. Merrill, F. Travostino, "DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks," *IEEE CCGRID/GAN 2004, Workshop on Grid and Advanced Networks*, April 2004.

[19] L. Smarr, A. Chien, T. DeFanti, J. Leigh, P. Papadopoulos, "The OptIPuter," *Special Issue: Blueprint for the Future of High Performance Networking Communications of the ACM*, Vol 46, No 11 Nov 2003, pp58-67.

[20] http://www.iwire.org

[21] http://www.east.isi.edu/projects/DRAGON/

[22] M. Blanchet, F. Parent, B. St. Arnaud, Optical BGP (OBGP): InterAS Lightpath Provisioning, *IETF Network Working Group Report*, March 2001. http://search.ietf.org/internet-drafts/draft-parent-obgp-01.txt

[23] http://www.canarie.ca/canet4

[24] http://www.doe.gov

[25] http://www.globusworld.org/program/slides/7c_1.pdf

[26] T. DeFanti, C. De Laat, J. Mambretti, K. Neggers, and B. St, Arnaud, "TransLight: A Global Scale LambdaGrid for E-Science," *Special Issue: Blueprint for the Future of High Performance Networking, Communications of the ACM*, Vol 46, No 11 Nov 2003 pp 34-41.

[27] http://www.glif.is

[28] http://www.startap.net/starlight

[29] http://www.surfnet.nl

[30] http://www.uklight.ac.uk

[31] http://www.teragrid.org

[32] http://www-unix.globus.org/toolkit

[33] "G.872: Architecture of optical transport networks," Telecommunication Standardization Sector (ITU-T) of the International Telecommunication Union (ITU), November 2001,

[34] S. Naiksatam, S. Figueira, S. Chiappari, and N. Bhatnagar, "Modeling Advanced Reservation Requests in Optical Networks," *COEN/SCU Technical Report*, June 2004.