DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks

S. Figueira[1], S. Naiksatam[1], H. Cohen[2], D. Cutrell[2], P. Daspit[2], D. Gutierrez[3], D. B. Hoang[4], T. Lavian[2], J. Mambretti[5], S. Merrill[2], F. Travostino[2].
[1]Santa Clara University, [2]Nortel Networks Labs, [3]Stanford University, [4]University of Technology, Sydney, [5]iCAIR and Northwestern University.
{sfigueira, snaiksatam}@scu.edu, {hcohen, dcutrell, pdaspit, tlavian, smerrill, travos}@nortelnetworks.com, degm@stanford.edu, dhoang@it.uts.edu.au, j-mambretti@northwestern.edu.

Abstract

Advances in Grid technology enable the deployment of data-intensive distributed applications, which require moving Terabytes or even Petabytes of data between data banks. The current underlying networks cannot provide dedicated links with adequate end-to-end sustained bandwidth to support the requirements of these Grid applications. DWDM-RAM[1] is a novel service-oriented architecture, which harnesses the enormous bandwidth potential of optical networks and demonstrates their on-demand usage on the OMNInet. Preliminary experiments suggest that dynamic optical networks, such as the OMNInet, are the ideal option for transferring such massive amounts of data. DWDM-RAM incorporates an OGSI/OGSA compliant service interface and will promote greater convergence between dynamic optical networks and data intensive Grid computing.

1. Introduction

DWDM-RAM is a DARPA [10] funded research experiment established to design and implement in prototype a new type of Grid service architecture that is optimized to support data intensive Grid applications through advanced optical networking. DWDM-RAM closely integrates Grid data services and optical networking services.

One of the goals of Grid architecture development is to enable efficient support for data-intensive applications, which may require moving - without prior notice - Terabytes or even Petabytes of data among multiple sites. One of the barriers to achieve this goal is the fact that traditional technology used in currently deployed networks limits at many levels the transfer of massive amounts of data. Although availability of sufficient bandwidth is an issue, access to resources that provides this bandwidth solves only one part of the problem. The complete solution is a new network architecture with the ability to orchestrate data flows with many different types of characteristics, including those requiring exceptionally high bandwidth for sustained periods of time. In the DWDM-RAM model, these high performance data flows are provided by dedicated optical paths, which are dynamically allocated. State of the art DWDM technology based on dynamic wavelength switching enables the creation of Grid services that allocate and release these paths either on-demand or by advance reservation.

Grid applications typically need to allocate and reserve multiple types of resources, such as computational, data, instrumentation, and networks, at distributed sites. Services such as the Globus Resource Allocation Manager (GRAM) job scheduler [8] have been developed to coordinate and schedule the computational and data resources needed by Grid applications. In [12], the authors propose GARA, the General Purpose Architecture for Reservation and Allocation, which enables the construction of application-level co-reservation and co-allocation libraries that applications can use to dynamically schedule collections of resources. In GARA, the network is treated as a primary resource, which can be allocated and reserved as well, and a service provides this resource.

To enable the efficient use of optical networks as a primary resource, a novel architecture for the reservation and dynamic allocation of individual

[1] DWDM-RAM research was made possible with support from DARPA, award No. F30602-98-2-0194.

wavelengths is required. This architecture relies on a new network service, which provides options for allocating dedicated network resources. This service must discover the network topology and both explore the availability and optimize the schedule of the optical network resources. It should also present a standardized, high-level and network-accessible interface. A natural choice for implementing this interface is the Open Grid Service Interface (OGSI) [27]. Such interfaces are compliant with the GGF's OGSA specification [13] and, in addition, conform to widely used Web Services standards (WSDL, SOAP, and XML).

In addition to presenting an OGSI-compliant interface, the network service should have a standard way of representing wavelength resources for communicating with clients. No such standard currently exists. For the Grid community, a promising approach would be to extend the XML form of the Resource Specification Language (RSL) [16]. This RSL schema is currently used by GRAM for other resources. Adding optical network extensions to RSL would make it possible to enhance GRAM to handle optical network resources.

This paper presents an implementation of the aforementioned concepts as a part of the DWDM-RAM research project. This project encompasses the development of two services, the Data Transfer Scheduling (DTS) service and the Network Resource Scheduling (NRS) service. The DTS service's goal is to provide scheduled data transfer between data banks. The NRS service's goal is to provide dynamic dedicated wavelength, both on demand and by advance reservation. The DTS uses the NRS service to provide the user application with an optimized utilization of the optical network. A preliminary implementation of this architecture was first demonstrated during the 9th Global Grid Forum, in Chicago, October 2003 and then at the Supercomputing Conference, in Phoenix, November 2003. Other results of experiments with dynamic wavelength on advanced optical networks provisioning for data-intensive Grid applications have been recently published [22].

This paper is organized as follows: Section 2 describes dynamic optical networks and their features, Section 3 introduces the DWDM-RAM architecture, Section 4 presents the interface created to allocate wavelengths dynamically, Section 5 describes the testbed used in the implementation and presents initial results, and Section 6 concludes and discusses future work.

2. Optical Networks

Bulk data transfer oriented grid computing requires quality of service, in particular, guaranteed minimum bandwidth and minimized packet loss, which are not easily achievable in packet switching networks. Recently, Grid developers have been using IntServ [6] and DiffServ [4] approaches to obtain QoS in packet switching networks ([11][12][15][21][25]). IntServ and DiffServ may help in some special cases, but they are far from perfect solutions. Optical networks have the potential to change the way networks are used in Grid computing. Optical networks based on wavelength switching can easily provide guaranteed bandwidth and performance in terms of low bit error rates. In fact, based on the same assumption, other groups have also been focusing on providing optical networks to the Grid community [26][29].

A main concern about wavelength switched optical networks is the overhead incurred during the end-to-end path setup. However, it should be noted that the time taken in the transfer of the huge amounts of data generated by Grid applications amortizes the time spent in the setup. The graph in Figure 1 illustrates this fact. It shows the amount of time spent in the path setup as a percentage of the total transfer time versus data transfers of different sizes. These numbers were generated synthetically in order to motivate the usage of optical networks in different scenarios. We indicate in the graph the threshold at which the setup time is about 5% of the total time, i.e., the overhead is amortized and becomes insignificant.

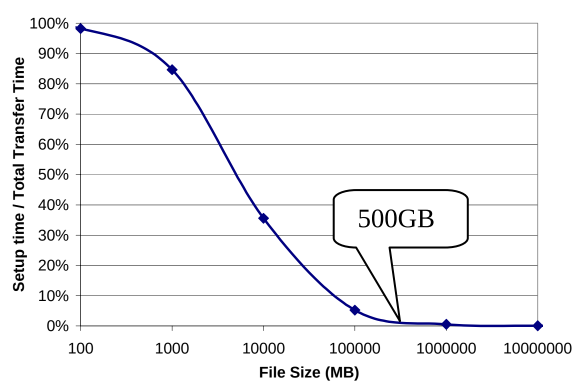


Figure 1: Overhead is insignificant at 500 GB

In Figure 1, we assume a setup time of 48 sec and a bandwidth of 920 Mbps, in which case the threshold is 500 GB. Since Grid applications usually transfer huge data sets, it is clear that the time for the path setup is negligible in Grid computing.

Note that 48 sec is the currently observed average upper-bound time for the path setup, and 920 Mbps is the currently observed maximum throughput (see Section 5). Of equal importance is the time required to deallocate resources to enable subsequent usage, once the data transfer is complete. This has been observed to be around 11sec (see Section 5) and is also insignificant in comparison to the data transfer time.

3. The DWDM-RAM Architecture

The DWDM-RAM architecture shown in Figure 2 closely fits the Grid Architecture described in [14]. This architecture also leverages the Globus Toolkit 3's functionality. The main modules are explained below.

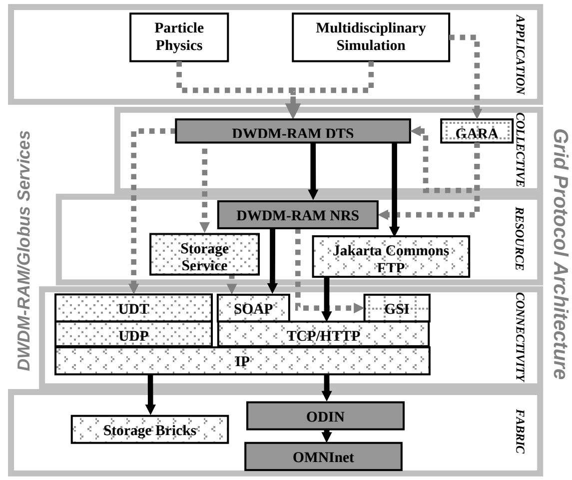
Fabric: OMNInet and ODIN

The OMNInet photonic testbed network [3] implemented across a metro area constitutes the shared resource in the DWDM-RAM architecture. Optical Dynamic Intelligent Network[2] (ODIN) [17][18][22] is the software suite that controls the OMNInet through various lower-level API calls presented in Section 5. ODIN has been designed for extremely high performance, long term Terabit data flows with flexible and fine grained control. It is a stateless server implementation, which includes a simple API to provide path provisioning and monitoring to the higher layers.

Connectivity: TCP/IP, HTTP, SOAP, JAX-RPC

Currently, the DWDM-RAM system uses standard off-the-shelf communication protocol suites to communicate requests and responses between application clients and DWDM-RAM services and also among DWDM-RAM components. Requests and responses conform to OGSI and Web Services specifications, i.e., they are SOAP messages, carried in HTTP envelopes and transported over TCP/IP connections. The SOAP messages are encoded using JAX-RPC to facilitate interoperability between existing Java-based Globus toolkit and the DWDM-RAM components.

[2] ODIN research was made possible with support from the National Science Foundation: award - ANI-0123399



	Core DWDM-RAM components		System specific components		Connects currently implemented components
	Globus Toolkit components		Future candidate components		Connects future candidate components

Figure 2: The DWDM-RAM architecture

Resource: DWDM-RAM Network Resource Scheduling (NRS) Service

The Resource and Connectivity protocol layers form the neck of our "hourglass model" [23] based architecture. The NRS is essentially a resource management service, which supports protocols that offer advance reservations and QoS. It maintains schedules and provisions resources in accordance with the schedule. It provides an OGSA/OGSI compliant interface to request the optical network resources. It has complete understanding of dynamic lightpath provisioning and communicates these requests to the Fabric layer entities. In our architecture, the NRS can stand alone without the DTS (see Collective layer). GARA integration is an important avenue for future development. A key feature is the ability to reschedule reservations, which satisfy under-constrained scheduling requests, in order to accommodate new requests and changes in resource availability. Requests may be under-constrained through specification of a target reservation window, which is larger than the requested duration, through open-ended specifications, such as "ASAP", or through more sophisticated request specifications, such as client-provided utility functions for evaluating scheduling costs. Models for priorities and accounting can also be incorporated into the scheduling algorithms.

Collective: DWDM-RAM Data Transfer Scheduling (DTS) Service

The DTS service is a direct extension of the NRS service. The DTS shares the same backend scheduling engine and resides on the same host. It provides a higher-level functionality that allows applications to schedule advance reservations for data transfers without the need to directly schedule path reservations with the NRS service. The DTS service requires additional infrastructure in the form of services living at the data transfer endpoints. These services perform the actual data transfer operations once the network resources are allocated. As with the NRS, the DTS has an OGSI interface, and GARA integration is a planned avenue for future development. As an extension of the NRS, the DTS also employs under-constrained scheduling requests in order to accommodate new requests and changes in resource availability.

It is important to note that, in the DWDM-RAM design, the application at the endpoints drives the state and decisions by DTS, NRS, and ODIN, thus in full compliance with the end-to-end argument [24]. These intervening layers never attempt to re-construct nor second-guess an application's intent from traffic. Instead, they follow the explicit command and control from the application. These layers are best positioned to appreciate both application's commands and network status, while bridging meaningfully between the two.

Also, we plan to have DWDM-RAM comply with the so-called fate-sharing principle [7]. This design principle argues that it is acceptable to lose the state information associated with an entity if, at the same time, the entity itself is lost. In this case, DTS, NRS, and ODIN will make a point of reclaiming resources once an application is terminated, or is found to be non-responsive for a configurable time interval.

4. Dynamically Allocating Bandwidth On-Demand

For wavelength switching to be useful for Grid applications, a network service with an application level interface is required to request, release and manage the underlying network resources. Previous work has been done in defining an OGSA/OGSI based network service interface to network resources. In [12], for example, the authors describe a service that provides allocation of bandwidth in IntServ networks. Our approach follows their model but develops a more comprehensive schedule and reservation based Network Resource Scheduling (NRS) service, which provides user applications with access to underlying optical network resources. This service should guarantee dedicated access to the optical links, which may be requested on-demand or by advance reservation. Advance reservation requests can be under-constrained, which means the request can be satisfied by more than one possible time slot. This allows the service to reschedule reservations as needed to satisfy future requests and changing conditions.

The NRS also attempts to provide an interface with different levels of detail. Some coordinating and scheduling clients may need only high-level facilities like those for individual applications. For these applications, the interface should allow the request of light paths, but these clients do not need to have knowledge of the details of the underlying network topology or management protocols. However, clients that attempt to optimize network resources use may need a richer interface, e.g., they may need to be able to schedule individual optical segments. For these clients, the interface should allow the request of individual segments. Although these clients may need knowledge of the network topology, they should still be insulated from the network management protocols.

We have started by implementing a network service with a simple application-level interface. This service is able to allocate dynamically dedicated end-to-end light paths requested on demand. The following were the goals for this initial interface:

- It should be usable by Grid applications.
- It should be network-accessible from any authorized host.
- It should be standards-based and not require any proprietary technology.
- It should allow the application to specify parameters of the desired optical path that are relevant at the application level.
- It should hide the details of the underlying network topology and network management protocols.
- It should support the immediate allocation of bandwidth, but should be easy to extend in order to incorporate future scheduling.

At the highest level, the interface described here is wrapped in a Java implementation that shields the caller from the details of the application-level protocols, which are needed to communicate with the service. The service itself hides various lower level network topology and management protocols from the caller.

This current interface exposes two methods to the user: `allocateNetworkPath` and `deallocateNetworkPath`. The method `allocateNetworkPath` requests a path between the end hosts. The path allocated should meet the criteria specified in the parameter object passed to the method. These parameters include the network addresses of the hosts to be connected, the minimum and maximum acceptable bandwidths, and the maximum duration of the allocation. The `allocateNetworkPath` method returns to the caller an object containing descriptive information on the path that was allocated. This object also serves as a handle for the `deallocateNetworkPath` request. By adding parameters representing earliest and latest acceptable start times, this interface can be extended to accommodate under-constrained advance reservations.

The NRS interface suggests likely properties for a service that may implement this interface. Such a service can be implemented in a way that does not maintain any session or context information for any particular client between calls. The only necessary context information is the allocated path identifier, which the client is required to supply in order to deallocate a path. The service must maintain this information about these allocated paths so, in this sense, it is not "stateless," but each client call can be treated as a self-contained unit and processed entirely in a single-message exchange. Thus, the interface fits the service oriented architecture of Web Services [5] quite closely.

At the same time, we believe that it is important to conform to the emerging Open Grid Services Architecture (OGSA) Grid standard to be accessible to Grid applications that use this architecture. Hence the NRS interface also conforms to the Open Grid Services Infrastructure (OGSI) interface for services. This interface, in its basic form, is a Web Service interface with some additional conventions from the OGSI standard. Although client access is via a normal Java interface as described above, internally, the client-service interface is an OGSI Web Service implemented using SOAP and the JAX-RPC API. The OGSI implementation is quite lean (no SDEs), and therefore the conversion to new specifications like the WS-Resource Framework (WS-RF) [30] will be a somewhat mechanical process. In fact, the DWDM-RAM architecture builds the concept of a Network Service from the ground up and is fairly well insulated from either OGSI, or WS-RF, or any other emerging specification that may prevail. In any case, the user of the NRS service is spared any direct contact with the underlying middleware.

5. Testbed, Experiments and Result

The testbed for the DWDM-RAM system is provided by a next-generation large-scale Metro/LAN dynamic optical network extent called the OMNInet [3], operational since 2001. OMNInet uses metro dark fiber infrastructure, which has been provided by SBC to connect locations sponsored by Northwestern University, and a mix of product and research equipment from Nortel Networks. OMNInet has been deployed at four locations in the Chicago metro area and is configured in a partial-mesh 10GE DWDM network. Each location has a MEMS based 8 x 8 DWDM photonic switch interfaced with a Nortel Passport 8600 Ethernet Routing Switch, providing 4 x 10GE line interfaces and up to 32 x 1GE client interfaces. The OMNInet configuration is shown in Figure 3.

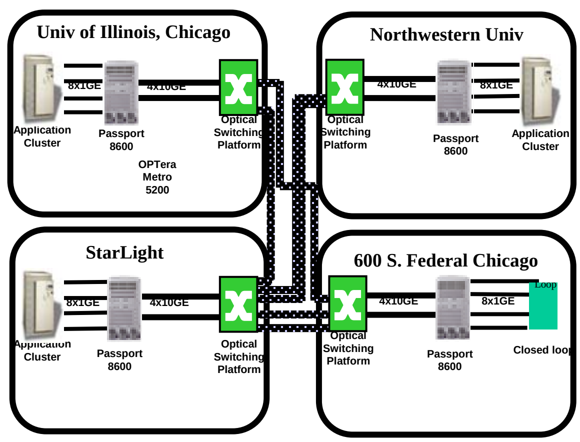


Figure 3: OMNInet core nodes

The dynamic light path provisioning of the OMNInet is provided by the Optical Dynamic Intelligent Network (ODIN) services module, which manages the optical network control plane and resource provisioning including path allocation, deletion, and setting of attributes. ODIN uses an OIF UNI 1.0 [28] compliant UNI library to interface with the OMNInet unified control plane. The control plane is orchestrated by a GMPLS [20] based CR-LDP [1] implementation, which is used to establish the label forwarding state along the routes computed by an enhanced OSPF (constraint-based routing) algorithm[5]. (Note: An RSVP-TE [2] implementation to support advance reservations is also in the process of being deployed.) The technological breakthrough in the design of the OMNInet photonic switches, coupled with the tight integration of the control software, brings down the provisioning time from months to a few seconds. This is the critical enabling factor in on-demand provisioning, which makes dynamic optical networks an important technology for data-intensive Grid computing.

Figure 4 compares the efficiency of packet switching with lambda switching, executing with different amounts of bandwidth, by plotting the amount of data transferred against the time taken in seconds. This graph was generated synthetically. The path setup takes 48 sec which was recorded in experiments (explained in detail later). The path setup time includes the time required to request a path, allocate the path, configure the layer 1 and 2 components of the network and update the relevant tables. It is important to note that only a miniscule fraction (approximately 20 milliseconds) of this delay is actually due to the setting up of the path within the optical switch. A part of the setup delay is also due to the propagation of state information through the photonic nodes and optical paths. 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, and with fairly simple optimizations, we expect to bring this 48 sec delay down significantly.

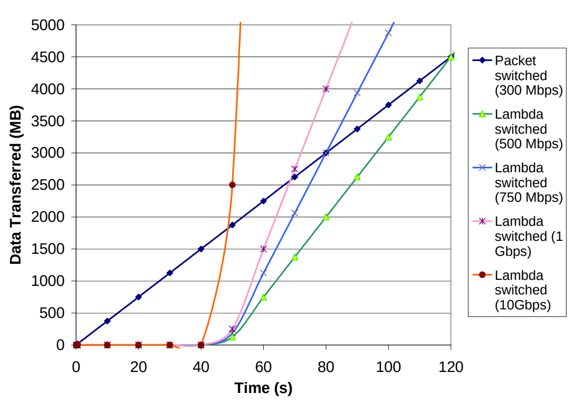


Figure 4: Packed Switched vs. Lambda Network -- Setup time tradeoffs.

[5] For the OMNInet metro photonic networking research, the GMPLS, O-UNI, LMP standards are implemented as Nortel Networks proprietary Wavelength Routing and Distribution Protocol (WRDP) optical control plane software suite.

The important information obtained by this graph is the crossover points at which the optical network becomes a better option than the packet switched network. Note that when the path setup takes 48 sec, the crossover point can be with files as small as 1.9 GB and up to 4.5 GB. The graph affirms the enormous advantage of the DWDM-RAM system over other packet switching systems when transferring large data sets and presents an exciting option for the Grid community.

Note that, for transferring smaller amounts of data, the packet switching traditional approach may be a better option. The appropriate network to use, in each case, should be decided by co-allocation services, as defined in [9][12]. These co-allocation scheduling services use information about the resource's capabilities and the application's requirements to allocate a performance-efficient set of resources for the application.

Figure 5 presents a throughput graph for a 30 GB memory-to-memory transfer. The smooth curve is the average effective data-transfer rate, including the startup cost of allocating the lightpath. The jagged curve, averaging around 440 Mbps, is the instantaneous throughput, measured in 0.1 sec intervals on the receiving machine. Note that the throughput periodically jumped to about 530 Mbps for a sustained period of about 100 seconds, a phenomenon under investigation. For this 30 GB data transfer, we can see that the startup costs begin to be amortized after about 230 sec or 12.2 GB.

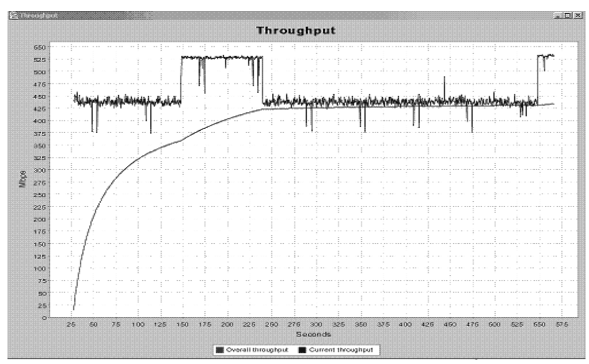


Figure 5: Instantaneous and average throughput for a 30 GB memory-to-memory transfer over OMNInet using TCP as transport protocol.

The above experiment was run on dual PIII 997MHz machines with 512MB memory, 1GB swap space, running on Red Hat Linux 7.3 (Kernel: 2.4.18-3smp) and using 1 GigE NIC cards. Note that we used the Linux networking protocol stack with no performance tuning.

When performing file transfer in the above setup we observed large oscillations in the instantaneous throughput, which resulted in significantly lower average throughput. We suspected that this was an artifact of using standard TCP. This prompted us to experiment using UDT [19], a high performance data transfer protocol. Figure 6 shows a 15 GB memory-to-memory data transfer using UDT as the underlying transport protocol. The throughput has been sampled at 0.1 sec intervals. The maximum and the average throughput recorded were 920 Mbps and 680 Mbps, respectively.

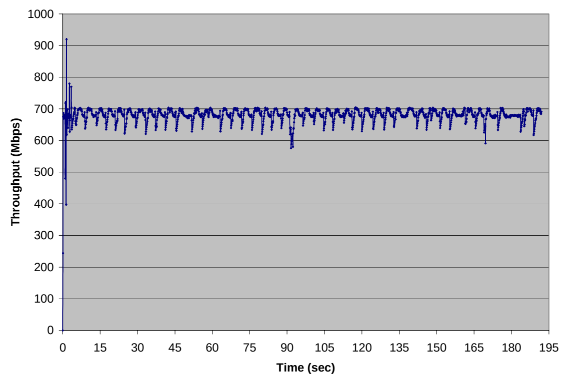


Figure 6: Instantaneous throughput recorded for a 15 GB memory-to-memory transfer over OMNInet using UDT as transport protocol.

The numbers in Figure 6 show a better performance than those in Figure 5, reinforcing our initial suspicion that TCP was affecting the throughput. We also performed file transfer experiments using UDT. For a 20 GB file transfer, the maximum and the average throughput recorded were 376 Mbps and 272 Mbps respectively. These are significantly lower than those seen in Figure 6, leading us to believe that issues like disk I/O on both ends of the data pipe, buffering, and similar factors impact the average throughput in the disk-to-disk file transfers.

Table 1 shows a breakup of the total end-to-end time required for the above file transfer operation to complete. It is important to note that the lightpath used to transfer data in the experiment described was dynamically allocated and dedicated to the file transfer.

While UDT does seem to have had a favorable impact on the average throughput, it is evident from the graph of Figure 6, that there are still oscillations in the instantaneous throughput. Also, the average throughput is less than the maximum 1 Gbps capacity of the channel. Currently, the default UDT protocol values are being used, namely, UDT buffer size of 40,960,000 bytes, UDP send buffer of 64KB, UDP receive buffer of 4 MB, and flow window size of 25600. We believe that tuning these parameters to the OMNInet testbed will yield a higher throughput.

Table 1: Application level measurements recorded for a 20GB file transfer

Event	**Seconds**
Start : File transfer request arrives	0.0
Path Allocation request	0.5
ODIN server processing	2.6
Path ID returned	0.5
Network reconfiguration	45
Data transfer	622
Path deallocation request	0.3
Path tear down	11.3

6. Conclusion

In summary, we have demonstrated that a dynamic optical network can be encapsulated as an important resource accessed through a Grid service, which is necessary for data-intensive applications. The DWDM-RAM system presented here addresses limitations of traditional implementations and facilitates on demand optical channel (lightpath) provisioning. We have also presented an OGSA/OGSI compliant interface developed to ease the usage of the OMNInet by Grid applications.

The interface presented meets the needs of many applications, but it should be extended to include features that may be necessary or desirable in other circumstances. One of our main goals is to make the interface GARA compliant and able to provide a comprehensive, re-schedulable and window-based, network service as discussed in Section 3.

We are currently exploring UDT and other high performance transport protocols which can help to achieve a uniform throughput close to the maximum limits. This, coupled with the facility to aggregate several channels to transfer one large file, will dramatically impact the data transfer time perceived by the user.

While a photonic switched network like OMNInet may be ideal for a DWDM-RAM-like architecture, our experiments suggest that DWDM-RAM could also be implemented on a traditional optical-electrical-optical (OEO) network. Providing a solid interface to such optical networks is a key step in enabling its usage by the Grid community. Our initial experiments have been extremely promising, and we are continuing to improve the network service.

Acknowledgements

The authors would like to thank Aaron Johnson, Fei Yeh, Jeremy Weinberger, Jim Chen, and David Lilithun from iCAIR, and Ramesh Durairaj, Inder Monga, and Phil Wang from Nortel Networks, for their invaluable contributions.

References

[1] P. Ashwood-Smith (Ed) and L. Berger (Ed), "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Label Distribution Protocol (CR-LDP) Extensions," RFC 3472, January 2003.

[2] D. Awduche et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, December 2001.

[3] E. Bernier et al., "OMNInet: A Metropolitan 10Gb/s DWDM Photonic Switched Network Trial," Submitted for OFC 2004, Feb 22 – 27, 2004. (In review)

[4] S. Blake et al., "An Architecture for Differentiated Services," IETF RFC 2475.

[5] D. Booth, et al., "Web Services Architecture," http://www.w3.org/TR/2003/WD-ws-arch-20030808/wsa.pdf

[6] R. Braden et al., RFC 1633: Integrated Services in the Internet Architecture: an Overview, *Internet RFC 1633*, July 1994.

[7] D. Clark, "The design philosophy of the DARPA Internet protocols," in *Proceedings of ACM SIGCOMM*, Stanford, CA, pp. 106--114, August 1988.

[8] K. Czajkowski et al., "A Resource Management Architecture for Metacomputing Systems," in *Proceedings of IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 62-82, 1998.

[9] K. Czajkowski et al., "Resource Co-Allocation in Computational Grids," *Proceedings of the Eighth IEEE International Symposium on High Performance Distributed Computing (HPDC-8)*, pp. 219-228, 1999.

[10] Department of Defense Advanced Research Projects Agency (http://www.darpa.mil/)

[11] I. Foster et al., "End-to-End Quality of Service for High-end Applications," *Computer Communications, Special Issue on Network Support for Grid Computing*, 2002.

[12] I. Foster et al., "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation," *International Workshop on Quality of Service*, 1999.

[13] I. Foster et al., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Open Grid Service Infrastructure WG, Global Grid Forum, June 2002.

[14] I. Foster et al., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International Journal of Supercomputer Applications*, 15(3), 2001.

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

[16] "The Globus Resource Specification Language RSL v1.0," http://www.globus.org/gram/rsl_spec1.html.

[17] R. Grossman et al., "Experimental Studies Using Photonic Data Services at iGrid 2002," *Journal of Future Computer Systems*, Elsevier Press, pp.945-956, August 2003.

[18] R. Grossman et al., "Photonic Data Services: Integrating Path, Network and Data Services to Support Next Generation Data Mining Applications," *NGDM'02, Proceedings*, November 2002.

[19] Y. Gu and R. Grossman, "End-to-End Congestion Control for High Performance Data Transfer," submitted to *IEEE/ACM Transactions on Networking*.

[20] L. Berger (Ed), "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description," RFC 3471, January 2003.

[21] G. Hoo et al., "QoS as Middleware: Bandwidth Reservation System Design," *Proceedings of the 8th IEEE Symposium on High Performance Distributed Computing*, pp. 345-346, 1999.

[22] J. Mambretti et al., "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.

[23] "Realizing the Information Future: The Internet and Beyond," National Academy Press, 1994, http://www.nap.edu/readingroom/books/rtif.

[24] J. Saltzer et al., "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, pp. 277-288, 1984.

[25] V. Sander et al., "A Differentiated Services Implementation for High-Performance TCP Flows," *Elsevier Computer Networks*, 34:915-929, 2000.

[26] L. Smarr et al., "The OptIPuter," *Communications of the ACM*, Vol. 46, No. 11, pp. 58-67, November 2003.

[27] S. Tuecke et al., "Open Grid Services Infrastructure (OGSI) Version 1.0," Global Grid Forum Draft Recommendation, June 2003.

[28] "User Network Interface (UNI) 1.0 Signaling Specification," http://www.oiforum.com/public/documents/OIF-UNI-01.0.pdf

[29] D. Vanderster, "The Return of Circuit Switching: A Review of User-Controlled Lightpaths for Grid Computing," Technical Report, University of Victoria, Canada, August 2003.

[30] "WS-Resource Framework," http://www-fp.globus.org/wsrf/