# Anycast Algorithms Supporting Optical Burst Switched Grid Networks

Marc De Leenheer[†], Farid Farahmand[*], Kejie Lu[$], Tao Zhang[**], Pieter Thysebaert[†], Bruno Volckaert[†],
Filip De Turck[†], Bart Dhoedt[†], Piet Demeester[†], and Jason P. Jue[‡]
[†]Department of Information Technology, Ghent University - IBBT - IMEC
[*]Department of Computer Electronics and Graphics Technology, Central Connecticut State University
[$]Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez
[**]Department of Computer Science, New York Institute of Technology Old Westbury
[‡] Department of Electrical Engineering and Computer Science, The University of Texas at Dallas
email: {marc.deleenheer@intec.ugent.be, farahmandfar@ccsu.edu}

***Abstract*— In this paper we consider implementing optical burst switching as a technology for building Grids with computationally intensive requirements. This architecture has been referred to as Grid-over-OBS (GoOBS). First, we briefly describe the proposed layered Grid architecture and show how OBS can be positioned within the Grid architecture. Then, we present a generic framework for anycast routing in the context of GoOBS when requests don't have an explicit destination address and they can be serviced by any appropriate Grid resource. We also develop several algorithms to support anycasting when only a single copy of a request is transmitted. Through simulation analysis, we show the performance of our anycast algorithms and compare them with traditional shortest-path unicast routing in which all jobs have specific addresses. Our performance analysis focuses on blocking probability of requests and average end-to-end delay.**

## I. Introduction

The astonishing advances in telecommunications and development of countless communication devices, has demanded massive computational power, data storage capacity, and networking capability. Such requirements have motivated researchers to develop the Grid. The Grid provides a practical and cost efficient infrastructure to accommodate scientific and business communities with their integrated computer-intensive requirements.

At the heart of the Grid is the network. Adequate networking allows geographically dispersed resources to be utilized collectively in order to satisfy a given application. Clearly, resource utilization of the Grid is limited by the available link bandwidth. Hence, integrating Grid resources with emerging high-performance optical network technologies, including optical switching and Dense Wavelength Division Multiplexing (DWDM), appears to be the natural choice [1]. A number of experimental testbeds, such as TransLight [2], have focused on developing such high-performance network elements for the Grid.

In general, the enabling technologies in the optical network infrastructure of the Grid, including the switching and resource allocation mechanisms, may be different depending on the Grid application. For example, a particular application may require moving large amounts of data (e.g. transferring multiple petabytes of astronomical data from multiple observation sites for analysis). For such applications, efficient and dynamic reservation of *lightpaths* is required at the Grid network level to guarantee sufficient bandwidth throughout the duration of the requests. A lightpath is typically defined as a dedicated end-to-end optical connection between two or more optical nodes. We refer to such a Grid-enabling architecture as Optical Circuit Switched (OCS)-based Grid or *Grid over Optical Circuit Switching* (GoOCS).

Many other Grid applications have computationally intensive requirements (e.g. mathematical problems requiring large number crunching). In fact, it is conceivable to imagine a number of users each with sub-wavelength bandwidth requirements but large processing power needs. In this case, the data is transmitted to suitable Grid resources and results are sent back to the clients after data processing has been completed. Such applications are often small in size, sensitive to latency, and require guarantee of service. Hence, satisfying them through establishing dedicated lightpaths, which include path setup and teardown and can take as many as tens of seconds, may not be efficient.

An alternative approach to meet computationally intensive Grid applications with moderate data size is to implement a new optical switching paradigm called Optical Burst Switching (OBS) [3]. In this architecture, referred to as *Grid over Optical Burst Switching* (GoOBS), one or more application requests, or *jobs*, are assembled into a super-size packet called *data burst*, which is then transported over the optical core network and forwarded to the appropriate Grid resources. Each data burst has an associated *control packet* containing information such as the burst's duration, source node, the type of Grid resources the burst requires, etc. Typically, the control packet is separated from the burst in space and time, i.e. transmitted on a dedicated control channel and apart from its associated burst by a time offset.

An attractive feature of GoOBS is its support of existing DWDM optical networking infrastructure and minimizing the need for optical-electrical converters at intermediate nodes. Another important advantage of GoOBS is its ability to utilize link bandwidth and Grid resources efficiently and provide low end-to-end latency. Such advantages have led a working group in the Global Grid Forum (GGF) to pursue the standardization of OBS in the context of Grid computing [4].

Fig. 1 shows the ratio of the signaling time, including the

0-7695-2622-5/06/$20.00 (c) 2006 IEEE

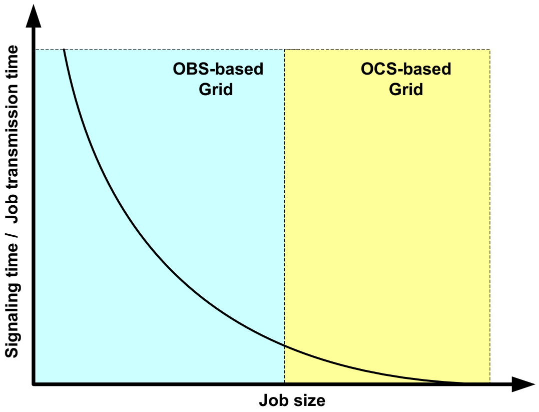

Fig. 1. The ratio of signaling time over total transmission time of a request (job) between the client and Grid resources as the job size varies.

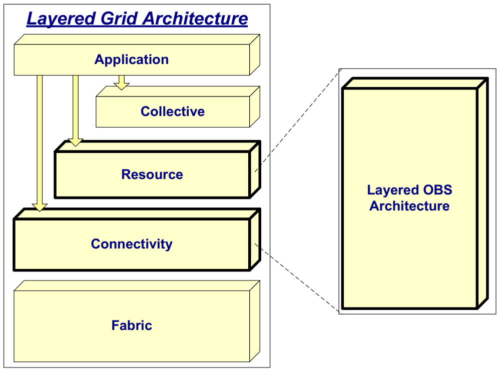

Fig. 2. A Grid-over-OBS architecture.

time required to setup and teardown lightpaths, over request (job) transmission time between the client and Grid resources as a function of job size. As demonstrated in this figure, if the ratio is reasonably small, say 5%, it is feasible to utilize OCS-based Grid. However, as the data size reduces and applications become more latency-sensitive, OBS-based Grid tends to be more efficient.

Implementing OBS as the transport mechanism for the Grid is a relatively new area and many important issues pertaining the GoOBS architecture are still uncovered. For example, it is not well understood how to aggregate multiple jobs in a single burst, how to retransmit a job in case of data burst loss, or how to route jobs to unspecified Grid resources in order to optimize the Grid's utilization; the latter issue is known as *anycast routing*.

GoOBS has been discussed previously in literature. In [4] the authors discuss solutions towards an efficient and intelligent network infrastructure for the Grid and propose taking advantage of recent developments in optical networking technologies, including OBS. Basic advantages of an OBS-based Grid are mentioned in [5], together with a discussion on the generic architecture. An OBS-like signaling protocol, called Just-In-Time, is introduced in [6] to enable optical networking for Grids.

In this paper, we position the OBS protocol within the framework of the layered Grid architecture. The main contribution of this paper is to present a generic framework for anycast routing in the context of GoOBS when jobs don't have explicit addresses and they can be serviced by any appropriate Grid resource. We develop several routing algorithms to support anycasting when only a single copy of a job is transmitted. Through simulation analysis, we show the performance of our anycast algorithms and compare them with the shortest-path unicast routing in which all jobs have specific addresses. This performance comparison will be based on average end-to-end delay and blocking probability of jobs.

The rest of this paper is organized as follows. In section II, we briefly review the general layered Grid architecture and how the OBS protocol can be positioned within the layered Grid architecture. In section III, we present a formal formulation of the anycast routing problem and introduce several anycast routing algorithms. Finally, Section IV discusses performance results obtained for the algorithms by means of simulations, followed by concluding remarks in Section V.

## II. Grid-Over-OBS Architecture (GoOBS)

In this section, we briefly review the proposed layered Grid architecture [7]. The Global Grid Forum (GGF) has considered a layering approach in developing such architectures, giving upper layers access to common lower-level functionality. The resulting layered Grid architecture, as proposed by GGF, is shown in Fig. 2. We briefly describe each layer from bottom to top and indicate its basic functionalities.

- *Fabric:* provides the underlying base structure including the storage systems, computers, networks, and system descriptors.
- *Connectivity:* defines core communication and the capabilities of resources. It also defines the authentication, authorization and delegation utilities of the users. Communication protocols enable the exchange of data between Fabric layer resources and includes transport, routing and naming.
- *Resource:* provides access to information and computation. This layer provides information about the state, performance, and structure of the Grid system.
- *Collective:* deals with interactions that are global in nature, such as resource discovery, brokering, system monitoring, etc. This layer also enables application-specific tasks, including archiving, checkpointing and management.
- *Application:* refers to the many different commercial, scientific and engineering applications requiring one or more resources such as computing power and data storage, which are provided by the Fabric layer.

As demonstrated in Fig. 2, OBS can be considered as the networking technology at the *lower layers* of the protocol model providing alternatives for the physical, data link, and network layers. In our model, the layered OBS is used to perform similar services and interfaces as the Connectivity and Resource layers of the Grid architecture. A comprehensive treatment of the layered OBS is provided in [8].

## III. Anycasting Routing Protocols in GoOBS

### A. Network Assumptions

A generic network architecture of GoOBS, including DWDM links, Grid edge nodes and its interfaces to Grid resources, and OBS core nodes is provided in [5]. We consider the following network assumptions: the network consists of $|N|$ nodes and $|L|$ links; each burst with unprocessed jobs has a maximum tolerable end-to-end delay (slack time, $T_{slack}$) upon processing. A network node or router consists of one or more ports, which form the interface through which data is sent and received.

An OBS-based Grid is fundamentally different from the traditional IP-centric OBS network in a number of ways. For example, in GoOBS processed jobs embedded in a burst must be returned to their original source nodes (clients). In addition, a burst can be discarded for (at least) *two* reasons: burst contention at an intermediate node *or* lack of sufficient Grid resources throughout the network within a predetermined time period (slack time). We refer to the latter as *burst starvation*.

Another fundamental difference is that unlike IP-centric OBS networks, unprocessed jobs in the Grid may be assigned no explicit destination address, as long as they are properly processed and returned to their clients. Consequently, instead of requiring shortest-path-based unicast routing protocols to transmit a burst to a specific destination node, GoOBS supports *deflection-based anycast* protocols as its underlying communication mechanism. In such protocols, a burst can be sent to any OBS node with appropriate Grid resources and intermediate nodes, which lack sufficient resources, simply *deflect* the burst to the next proper hop.

### B. Problem Formulation

IP-based anycasting has been considered and discussed previously in literature, including [9]. Using the same basic concept, we informally define anycasting in the context of GoOBS as follows: a client transmits a job to an anycast address and the OBS network is responsible to provide best-effort delivery of the job to at least one, and preferably only one, of the suitable Grid resources accepting requests for that anycast address. It is, hence, evident that, unicast and multicast routing are both *special* cases of anycast routing.

The following formulation can be derived for GoOBS anycasting. *Assuming* the entire GoOBS network (including the physical topology, full routing knowledge, and all available Grid resources associated with each core node) is known; *given* a burst $B(s, D)$ with source node $s$, $s \in N$, and all possible nodes $D = \{r_1, r_2, ..., r_d\}$, where $D \subseteq N$, with available Grid resources; *find* a subset $r$ of $D$, to which we should send copies of the burst *and* the routing policy, which indicates how to route the burst to the set of destinations specified by subset $r$, such that the probability of the burst (jobs) being processed by at least one of the $r$ nodes is maximized, *subject* to burst's slack time.

In the above formulation, subset $r$, called the *anycast destination group*, can be variable or fixed. When the anycast destination group is variable, the burst is directed to nodes defined by $r$, however, any other node in set $D$ is also allowed to process the burst. On the other hand, when the anycast destination group is fixed, only the nodes specified in $r$ are allowed to process the burst.

Furthermore, depending on the size of the anycast destination group $|r|$, a number of different anycast protocols can be considered:

- *Single-copy anycast, ($|r| \leq 1$):* A single copy of the data burst is transmitted by the source;
- *Multiple-copy anycast, ($|r| > 1$):* Multiple copies of the data burst are sent to multiple nodes with proper resources.

Clearly, when multiple number of bursts are generated, care must be taken to avoid looping and unintentional processing of the same burst by multiple nodes. In this paper, we only focus on single-copy anycast, where $|r| \leq 1$ and subset $r$ can be variable or fixed.

An important aspect of the aforementioned anycast problem is the routing policy. Upon selection of the anycast destination group and its size, it is necessary to establish a routing mechanism in which bursts with embedded jobs can reach nodes with sufficient Grid resources. In this paper, we assume shortest path routing is available when destination nodes are specified. Furthermore, alternative deflection policies may also may be allowed.

### C. Anycasting Algorithm Description

In this subsection, we propose a number of heuristic anycast routing algorithms. These algorithms differ in the way they perform the following two basic operations: *destination assignment* and *burst deflection*. The functionalities of each operation vary depending on burst type and whether nodes are stateful or stateless, that is if network status is communicated between nodes or not, respectively.

We consider three distinct burst destination assignment schemes performed by the source node.

- *Soft assignment (SA):* The source selects a destination node with available Grid resources for the burst. This selection can be random or according to some weighted function. The assigned *soft* destination can be altered by other nodes due to contention or starvation. In addition, an intermediate node can accept a burst with a different soft destination, if the node has sufficient processing resources. SA is an example of the case where $r$ is variable and $|r| = 1$, indicating that besides the node assigned by $r$, any node in $D$ is allowed to process the burst.
- *Hard assignment (HA):* This is similar to SA, however, the assigned destination node by the source *cannot* be altered by any intermediate node. Note that HA is basically unicasting, which is considered to be a special case of anycasting. Clearly, HA constitutes the case where $r$ is fixed and $|r| = 1$.
- *No assignment (NA):* The source assigns no explicit destination node to the outgoing burst and simply hands over the burst (containing one or more jobs) to the network. Therefore, the burst will wander in the network until it finds appropriate Grid resources. If the slack time of the burst is expired, the burst will be discarded. When the burst arrives at an intermediate node, the node checks its available resources and if sufficient resources are not available, the burst is

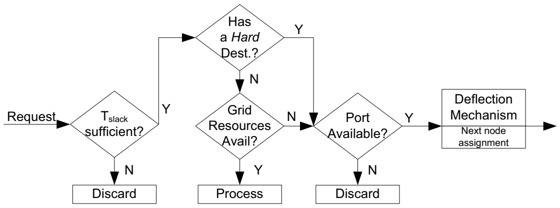

Fig. 3. Basic steps taken by network router in burst deflection operation.

forwarded to the next selected hop. NA is an example of the case where $|r| = 0$ and any node in $D$ can process the burst.

The basic motivation for implementing NA is to ensure that job processing throughout the network is not restricted to particular destination node(s). This is particularly important when some nodes appear to be overloaded. Hence, the NA mechanism must be implemented based on the assumption that the network has prior knowledge regarding the availability of its resources and it is able to intelligently direct a burst to any one of the possible destinations without guidance from the source node.

Let us now examine the burst routing and deflection policies. In general, burst deflection operation can be triggered at an intermediate or destination node due to contention or lack of sufficient processing resources, respectively.

Fig. 3 abstracts the general treatment of an incoming burst by an intermediate node. Note that the burst is initially checked for its slack time, $T_{slack}$, to ensure the burst is valid.

An important issue in burst deflection in case of contention is determining *where* to deflect the burst to. We consider three burst deflection schemes according to different resource availability criteria:

- *Random port availability (RPD):* In this case, upon contention, the burst is deflected to an available and randomly selected egress port. This scheme is similar to the hot-potato protocol in the sense that the node forwards the burst to the first available channel on any randomly selected egress port.
- *Weighted port availability (WPD):* This is very similar to RPD, except the port selection is based on some weighted function. Such function can include, for example, the port's blocking probability, whether the port is on an alternative shortest path to the original destination, etc.
- *Weighted Grid-resource availability (WGD):* In this case the node examines all available Grid resources throughout the network. Then, according to a weighted function, the node decides which egress port should be selected in order to forward the contending burst. The weight function can be *shifted* in favor of the ports providing alternative shortest paths to the original destination node.

Using the above framework, we consider a number of algorithms and describe their details below. We emphasize that our motivation in selecting these algorithms is to focus on anycasting and comparing its performance with the traditional shortest-path-based unicast algorithms.

We consider two sets of algorithms depending on whether deflection is allowed or not. We first describe algorithms with no deflection capacity.

*Soft destination assignment with no deflection (SA-ND):* In this case, a randomly selected destination is assigned to each outgoing burst and the burst will be routed on its shortest path toward the assigned destination. However, the burst can be processed by the first node with available resources along the shortest path. If the burst reaches its destination node and no processing resources were available, a new soft destination will be assigned.

*Hard destination assignment with no deflection (HA-ND):* In this case, each burst has a randomly assigned destination and it is forwarded along the shortest path to the assigned destination. If the assigned destination did not have sufficient resources to process the jobs embedded in the burst, a new destination will be assigned to the burst. Note that HA-ND is equivalent to the traditional shortest path based unicast routing algorithm.

*No destination assignment with no deflection (NA-ND):* In this case, we assume that no burst has an assigned destination, as in NA. Upon arrival at an intermediate node, the burst is processed if the node has sufficient capacity. If not, the burst is randomly assigned to an egress port which may or may not be available. If the selected port is not available, the burst will be dropped. The motivation for studying this algorithm is two-fold: to ensure that the load is properly balanced throughout all the egress ports at each node; and to use NA-ND as a baseline to study other variations of anycasting algorithms where no explicit destinations are assigned.

Next, we describe algorithms which support deflection. Depending on the deflection mechanism, we consider three different variations of the NA anycasting algorithm:

*No destination assignment with random port deflection (NA-RPD):* This is similar to NA-ND. However, in case the first randomly selected egress port was not available, the burst can be deflected to another randomly selected egress port on the node. The selection will continue until an available port is found. If no such port was found, the burst will be discarded.

*No destination assignment with weighted port deflection (NA-WPD):* In this case, if the first selected egress port is busy, the node will assign an alternative egress port. The port selection is based on finding the least congested egress port having the lowest measured blocking probability.

*No destination assignment with weighted Grid-resource availability deflection (NA-WGD):* In this case, when contention occurs at node $i$ and the first selected egress port is no longer available, the node must find an alternative egress port. This is performed by calculating the *Grid-resource availability function*, $\Gamma_p$, for each remaining port $p$:

$$\Gamma_p = \Sigma_{j,j \neq i} \frac{\Omega_j}{H_p(i,j)}, \tag{1}$$

In this equation, $\Omega_j$ is the available Grid-resources of node $j$, which has *not* been visited by the contending burst; $H_p(i,j)$ equals the number of hops of the shortest path from node $j$ to node $i$ through port $p$. If there is no path between node pair $(i,j)$, or such a path is not the shortest path through port $p$, $H_p(i,j)$ will be set to infinity. Using the above function, the alternative port

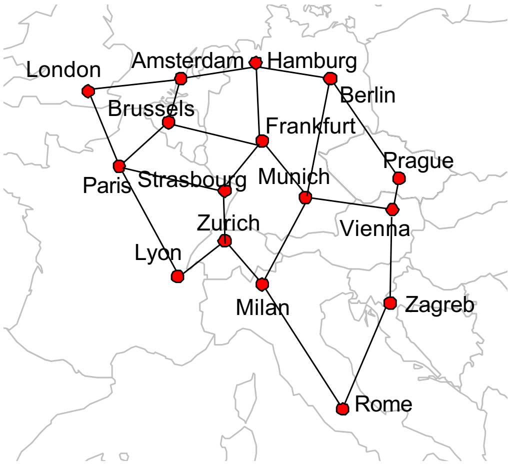

Fig. 4. European core network.

will be the one with the largest Γ value. Intuitively, this weighing function will direct a job towards the network region containing the nearest resources with the largest available capacity.

## IV. Performance Results

In this section, we present the simulation results obtained by implementing the aforementioned six algorithms. We consider the European core network (Fig. 4), containing 16 nodes and 23 bidirectional links. We assume all ports have 1 wavelength each operating at 40 Gbps, and Horizon [10] was implemented as wavelength reservation technology. Furthermore, 4 nodes were randomly selected to function as computational Grid resource, capable of processing a limited number of jobs (at most 50) in parallel. We consider both Poisson job arrivals and Pareto distributed interarrival times (Hurst parameter H = 0.9) at each network node, and assume each job is converted into a single optical burst. Both job data sizes and job execution times are exponentially distributed. The former has a mean of 1 MB, while the latter is initialised to generate a combined load of 80% on the computational resources, unless specified otherwise. In the following figures, one unit of network load (1 Erlang) equals one hour of network traffic collectively generated by all clients. Finally, the slack time is enforced by limiting each job to travel at most 10 hops in the network. Our results are focused on two performance metrics: the job blocking probability and average job hop count.

### *A. Destination Assignment*

Fig. 5 compares the blocking probability of jobs obtained for the different destination assignment mechanisms when no deflection is implemented (NA-ND, SA-ND and HA-ND) and as the network load varies. An interesting observation is that the performance of both NA-ND and SA-ND is much higher than HA-ND. This is because hard destination assignment is not sufficiently adaptive to the dynamic behaviour of Grid resources. Also note that SA-ND consistently outperforms NA-ND, since the inclusion of a soft destination increases the probability of a burst to reach a suitable resource. The generation of bursty traffic (Pareto) causes a consistent increase in blocking probability when compared to non-bursty traffic (Poisson).

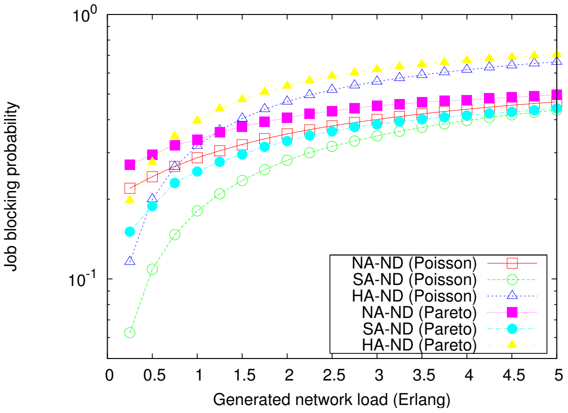

Fig. 5. Job blocking probability versus generated network load using different destination assignment techniques without deflection.

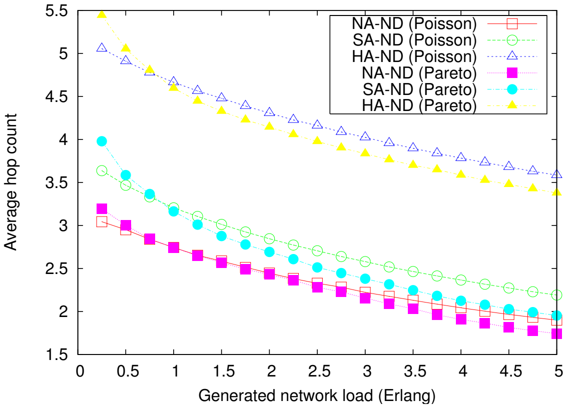

Fig. 6. Job average hop count versus generated network load using different destination assignment techniques without deflection.

The average hop count obtained by implementing NA-ND, SA-ND, and HA-ND for varying network loads is shown in Fig. 6. The lowest average hop count is achieved by NA-ND, which realizes a considerable improvement over the case when bursts are given specific destination nodes.

Fig. 7 shows the job blocking probability for different values of generated resource load, when the total resource capacity and the generated network load remain constant (2.5 Erlang). As before, HA-ND is outperformed by both NA-ND and SA-ND for the same reason given above. However, the initial improvement in performance of SA-ND over NA-ND is overturned as computational resources become continuosly overloaded (> 140%). The reason for this is provided by Fig. 8, which shows a much sharper increase in the average hop count for SA-ND than for NA-ND. The resulting increase in network utilisation of SA-ND consequently leads to a higher job blocking probability in the network.

### *B. Deflection*

Next, we examine the performance of the NA anycasting algorithm. Fig. 9 shows the blocking probability of NA with

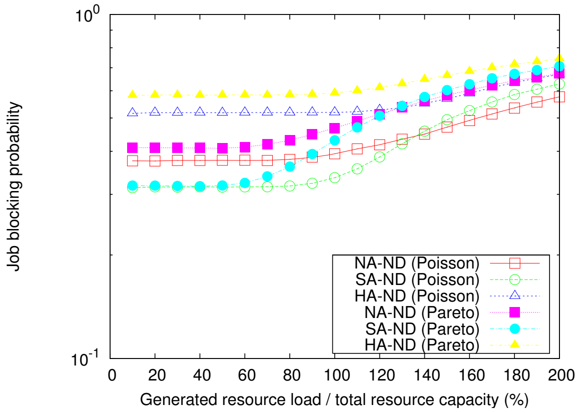

Fig. 7. Job blocking probability versus generated resource load using different destination assignment techniques without deflection.

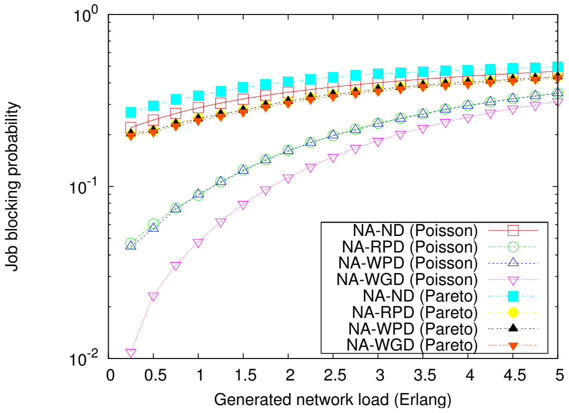

Fig. 9. Job blocking probability versus generated network load for no-destination assignment using different deflection mechanisms.

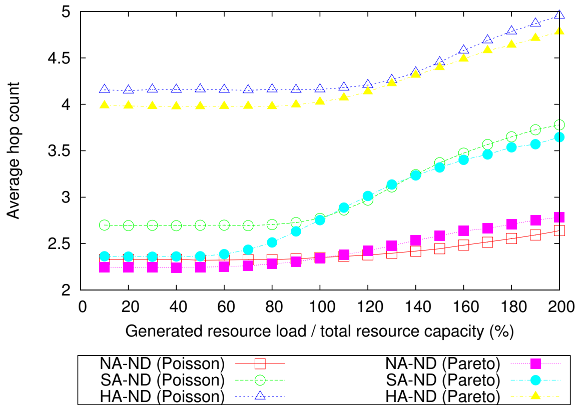

Fig. 8. Job average hop count versus generated resource load using different destination assignment techniques without deflection.

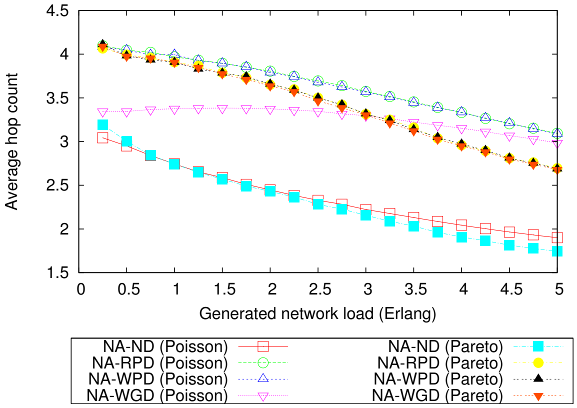

Fig. 10. Job average hop count versus generated network load for no-destination assignment using different deflection mechanisms.

different deflection mechanisms (NA-ND, NA-RPD, NA-WPD, and NA-WGD) for varying network loads. Our results indicate that NA-WGD results in the lowest blocking probability. Also, Fig. 9 demonstrates that the performance of NA-RPD and NA-WPD is very similar, in spite of the fact that NA-WPD is more complex in terms of hardware implementation because it requires maintaining port statistics.

The main drawback of deflection is that it increases the average hop count, as shown in Fig. 10. Note that NA-WGD appears to be a good tradeoff between job blocking and average hop count.

A final observation is the similarity between results for Poisson and Pareto job arrivals. Even though bursty traffic generally decreases performance when compared to a Poisson arrival process, the behaviour of the algorithms remains consistent over all presented alternatives.

## V. Conclusion

In this paper, the concept of a layered OBS network to support future Grids, has been presented. In this context, a generic framework for anycast routing was introduced. The gain in performance when jobs are given flexible (soft) destinations was demonstrated through simulation for different network and resource loads. Finally, a novel deflection technique, incorporating both network and Grid state information, was introduced. Simulation was used to show the improved performance over more traditional deflection techniques.

## References

[1] J. Mambretti et al. “The Photonic TeraStream: Enabling Next Generation Applications Through Intelligent Optical Networking at iGrid 2002”, *Journal of Future Generation Computer Systems*, 19(6), Aug 2003.

[2] T. DeFanti et al., “TransLight: A Global Scale LambdaGrid for E-Science”, *Communications of the ACM*, 46(11), Nov 2003.

[3] C. Qiao and M. Yoo, “Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet”, *Journal of High Speed Networks*, 8(1), Jan 1999.

[4] D. Simeonidou and R. Nejabati (editors), “Grid Optical Burst Switched Networks (GOBS)”, Global Grid Forum Draft, Jan 2006.

[5] M. De Leenheer et al., “A View on Enabling Consumer-Oriented Grids through Optical Burst Switching”, IEEE Communications Magazine, 44(3), Mar 2006.

[6] S. Thorpe et al., “Using Just-in-Time to Enable Optical Networking for Grids”, Proc. of the Workshop on Networks for Grid Applications, Oct 2004.

[7] I. Foster, “The Grid: A New Infrastructure for 21st Century Science”, *Physics Today*, 54(2), 2002.

[8] F. Farahmand et al., “A Layered Architecture for Supporting Optical Burst Switching”, Proc. of Telecommunications 2005, Lisbon, Portugal, July 2005.

[9] C. Partridge et al., “Host Anycasting Service”, RFC1546, Nov 1993.

[10] J. Turner, “Terabit Burst Switching”, *Journal of High Speed Networks*, 8(1), Jan 1999.

0-7695-2622-5/06/$20.00 (c) 2006 IEEE