Design and Control of Optical Grid Networks

(Invited Paper)

Marc De Leenheer, Chris Develder, Tim Stevens, Bart Dhoedt, Mario Pickavet, Piet Demeester
Dept. of Information Technology - IBBT
Ghent University
Gaston Crommenlaan 8 bus 201, 9050 Gent, Belgium
Email: {marc.deleenheer, chris.develder}@intec.ugent.be

***Abstract*—Grid computing aims to realize a high-performance computing environment, while increasing the usage efficiency of installed resources. This puts considerable constraints on the network technology, and ultimately has led to the development of Grids over optical networks. In this paper, we investigate the fundamental question of how to optimize the performance of such Grid networks. We start with an analysis of different architectural approaches (and their respective technological choices) to integrate Grid computing with optical networks. This results in models and algorithms to design optical Grid networks, and we show the importance to combine both dimensioning (offline) and scheduling (online) in the design phase of such systems. Finally, the concept of anycast routing is introduced and motivated. Both exact and heuristic algorithms are proposed, and their performance in terms of blocking probability and latency is presented.**

I. Introduction

Today, the need for network systems to support storage and computing services for scientific and business communities, are often answered by relatively isolated islands, usually known as clusters. Migration to truly distributed and integrated applications requires optimization and (re)design of the underlying network technology. This is exactly what Grid networks promise to offer: a platform for the cost and resource efficient delivery of network services to execute tasks with high data rates, processing power and data storage requirements, between geographically widely distributed users. Realisation of this vision requires integration of Grid logic into the network layers. Given the high data rates involved, optical networks offer an undeniable potential for the Grid. An answer to the demand for fast and dynamic network connections lies in the (relatively) new switching concepts such as Optical Packet Switching (OPS) and Optical Burst Switching (OBS [1]). Interest in optical Grid networks is being confirmed by the Open Grid Forum (OGF), a community of users, developers and vendors commited to the standardization of Grid computing. For instance, novel network paradigms and solutions to support OBS-based Grid networks, are presented in [2].

A major issue in the realization of high capacity optical networks, are the software tools and frameworks necessary for end-to-end, on-demand provisioning of network services. These need to be developed and refined to support coordination with other resources (CPU and storage) and will span accross multiple administrative and network technology domains. In response to the above requirements, the European IST project Phosphorus [3] is addressing some of the key technical challenges to enable on-demand, end-to-end network services across multiple domains. The Phosphorus network concept and testbed will make applications aware of the Grid environment, i.e. the state and capabilities of both computational and network resources. Based on this information, it is possible to make dynamic, adaptive and optimized use of heterogeneous network infrastructures connecting various high-end resources. The testbed will involve European NRENs[1] and national testbeds, as well as international resources (GEANT2, Internet2, Canarie, Cross Border Dark Fibre infrastructures and GLIF virtual facility). Finally, a set of highly demanding applications will be adapted to prove the concept.

Delivering the Grid promise implies answering a series of fundamental questions [4]: (re)design the architecture of a flexible optical layer, development of the necessary design techniques for e.g. dimensioning, and finally algorithms for routing and control, offering both QoS [5] and resilience guarantees. It is this to a large extent unexplored area of fundamental research that will be discussed in the following.

The remainder of this paper is structured as follows. First we discuss components and technologies in Section II, detailing the different network layers involved and the network scenario. In the same section, we also present a network performance model. We then proceed to the dimensioning of optical Grid networks (Section III), outlining two approaches to this problem. Finally, we discuss algorithms for realizing anycast routing in Section IV, after which our conclusions are presented.

II. Components and Technologies for Optical Grids

A. Network layers

A fairly generic view of a Grid is sketched in Figure 1. Users submit jobs to the network through a Grid User Network Interface (GUNI), thus providing the jobs' characteristics (processing, storage, priority/policy requirements, etc.). Likewise, Grid resources announce their capabilities (storage space, processing power, etc.) through a Grid Resource Network Interface (GRNI). Note that also the network characteristics, such as topology and bandwidth will need to be known to the

[1] National Research and Education Networks

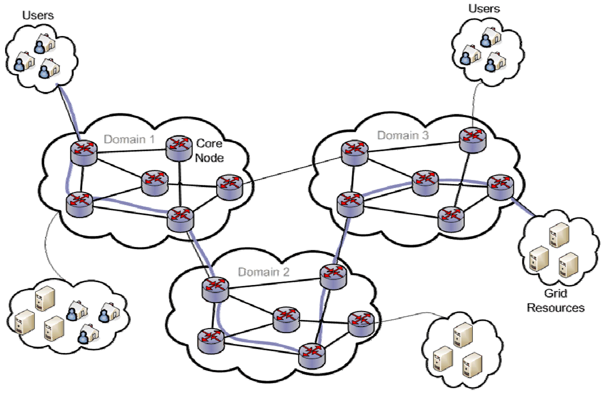

Fig. 1. Grid network infrastructure

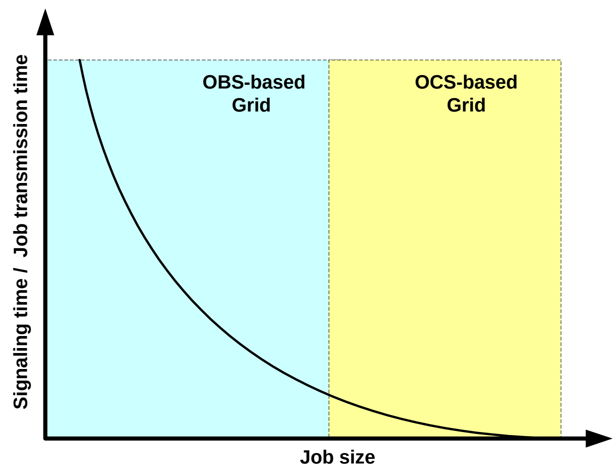

Fig. 2. Signaling overhead ratio

Grid scheduling and/or routing algorithms. The latter will be discussed in more detail in subsequent sections.

That optical technology can provide significant leverage for Grid networks is irrefutable, but whether to adopt an Optical Cricuit Switching (OCS) or rather an OPS/OBS paradigm is still debatable [6]. The main disadvantage of OCS is the signaling overhead involved. Depending on the ratio signaling time/job transmission time, OCS can be acceptable [7]: only if jobs require sufficiently long data transmissions (hence lightpath holding times are long compared to the setup and tear-down process), OCS makes sense. For small jobs, some form of rather complex grooming/aggregation at the OCS edges will be required to warrant efficient use of light paths. The qualitative Figure 2 indicates that, as job data size reduces and/or latency-sensitivity increases, OBS will be more efficient. Another advantage of a packet switching paradigm such as OBS is its ease in dealing with highly dynamic traffic patterns (both in space and time).

Where the aforementioned applies to OBS in general, an OBS-based Grid differs fundamentally from more conventional IP-centric OBS:

- The anycast routing paradigm: A Grid job does not care where it is executed (note that this does not apply to the jobs' processed results, which should be sent back to the job submitter). This topic will be further discussed in Section IV.
- Burst starvation: Bursts can not only be lost because of network contention (eg. no available wavelengths), but also through lack of Grid resources (CPU, disk space), preventing timely execution of a job. Refer to Section II-B for further details about this property.
- Future reservation: Jobs may be announced relatively long in advance. This notion of reservations of resources is not present in purely IP-based OBS.

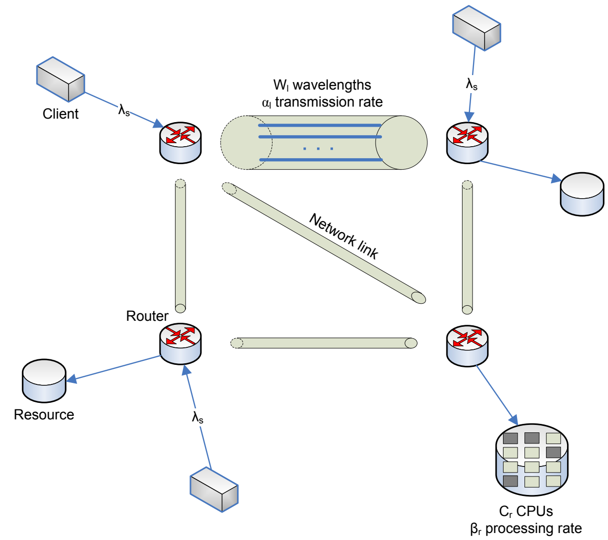

Fig. 3. Overview of Grid network model

When the Grid has to deal with a very heterogeneous population of jobs, it is conceivable to deploy a hybrid OBS/OCS architecture, where lightpaths are reserved for long-lasting jobs, while adopting OBS transmission for smaller ones [8]. A possible way of achieving OBS/OCS integration is through an ORION architecture, where gaps in wavelength usage are filled with easily extractable packets/bursts in so-called overspill mode [9]. Other specific sample OBS Grid architectures can be found in [10], [4].

We conclude this section on network layering with a brief discussion on the architecture of the basic building blocks of the Grid network, i.e. the core network routers. Obviously, the choices in switching approach are reflected in the design of optical routers, since these need to support user-defined bandwidth reservations for emerging applications over wavelength channels (circuit), optical bursts or even optical packets. The design of such a router is discussed in e.g. [11], where an architecture that combines slow and fast switching fabrics is proposed. The main advantage is the improved scaling behaviour of the switch, while offering the required flexibility of bandwidth demands.

B. Network Performance Model

The model presented in [12] accurately captures the characteristics of both network and resources present in an OBS-based Grid network. The solution technique used allows much greater scalability than simulation-based analysis would be able to achieve. This section presents the main concepts and results of this performance model.

The actual decision of where to process the burst and how to reach that destination, is traditionally made in scheduling entities. This decision is based on the current Grid state, the specific job requirements and various pre-determined optimization criteria. This approach has proven sufficient for most scenarios, but is not well adapted to the possible highly dynamic nature of a Grid environment. Indeed, in case large user groups are to be supported (e.g. consumer grids), the unpredictable and highly dynamic behaviour of user requests (and correspondingly, the resource and network states) can result in non-optimal use of existing infrastructure. A possible solution lies in the realization that there usually exist multiple, feasible resources for the execution of a specific job. As such, the assignation of a fixed, *hard destination* for a job should be abandoned in favour of a *soft destination*. Even though a job should still try to reach that soft destination, any suitable resource which is passed during the transfer, should be considered as a possible location for processing that job. Soft destination assignment can thus be regarded as an approach to schedule a job at multiple resources at once, whereby the availability of each resource is checked in a sequential manner. The soft destination approach can also be viewed as a form of *anycast routing* [13], since clients are not aware of which resource will do the actual servicing of the job. Finally, note that this mode of operation requires explicit support of the network's control plane; in the soft destination approach, the network router will be made aware of the resource availability. In this way, the router can quickly decide whether a specific job can be executed on the locally attached resource, instead of offering the job to the local scheduler and await its scheduling decision.

As shown in Figure 3, consider a network composed of a set L of directed links[2], each link l having W_l wavelengths with transmission rate α_l (expressed in jobs per time unit). Each link is terminated at both ends by a router, which are all capable of full wavelength conversion. The network also contains a set of sources S, with each source s generating jobs according to a Poisson arrival process with mean job arrival rate λ_s. Jobs are executed on a set of resources R, each resource r composed of C_r CPUs which have a mean processing rate β_r (jobs per time unit). Finally, each source and resource are connected to a single network router and their access link is neglected in this model (i.e. no blocking occurs on the access links).

Scheduling and routing policies are incorporated as follows. Let d_{sr} be the probability that a job which originated at source s is sent to resource r. This probability represents the scheduling policy (also referred to as destination assignment) of a source, and obviously, for each source s it holds that $\sum_r d_{sr} = 1$. The single routing path between each (source, resource) pair is represented by $P(s, r)$, which equals an ordered set of links. We assume that destination assignment follows a uniform distribution, i.e. each source sends an equal fraction of jobs to all resources. Additionally, shortest path routing is used.

[2]shown undirected in the figure for clarity

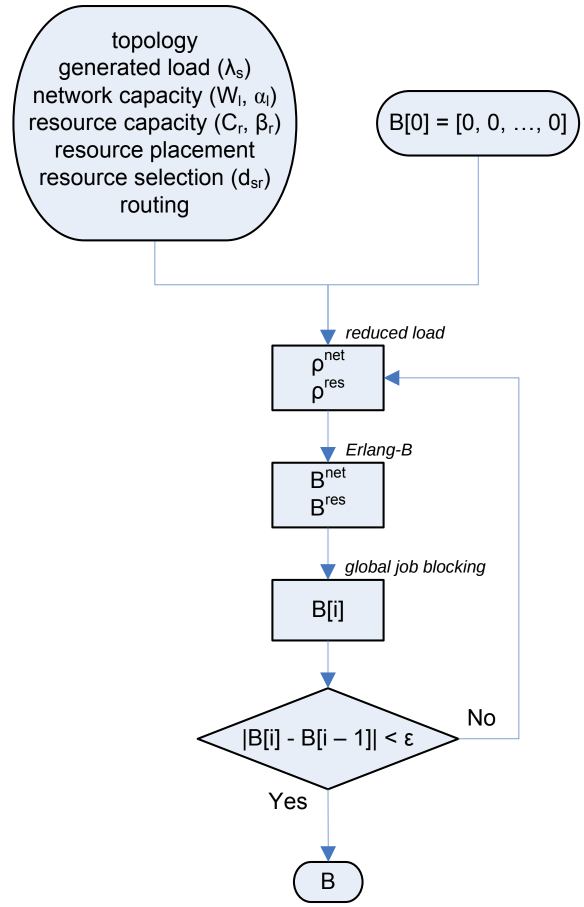

Fig. 4. Overview of reduced load approximation

Figure 4 shows an overview of our model and the different calculation steps. In general, we start from a given topology, the location and properties of clients and resources, and the implemented scheduling and routing policy. Based on this information, we want to obtain an estimate for the blocking probability of jobs in the Grid network. It is important to note that blocking can occur at two distinct locations in the network:

- in network links, due to network congestion, or
- in resources, due to overloaded resources.

To incorporate these two different causes for blocking, the algorithm starts by estimating the load on individual network links (ρ_l^{net}) and resources (ρ_r^{res}), based on a reduced load approach. This implies that the load on a network link or resource is reduced because of blocking events on other network links and/or resources. Consequently,we can calculate the individual blocking probabilities (B_l^{net} and B_r^{res}) by using the Erlang-B formula. This is based on the assumption that jobs are generated following a Poisson process, but this

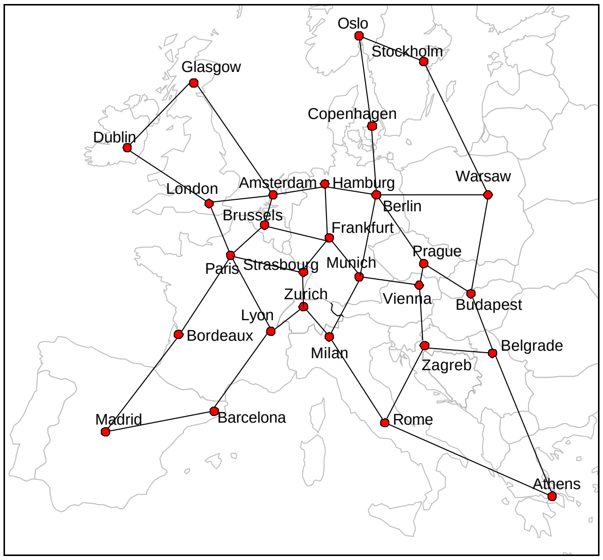

Fig. 5. Simulated topology: basic European network (28 nodes, 41 bidirectional links)

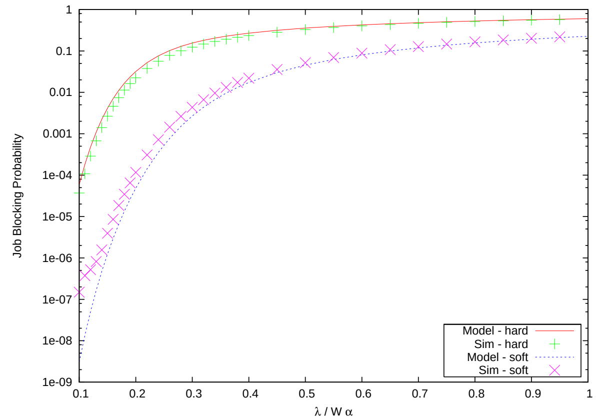

Fig. 6. Job blocking probability for varying generated network load and fixed mean generated resource load ($\frac{\lambda}{C\beta} = .01$)

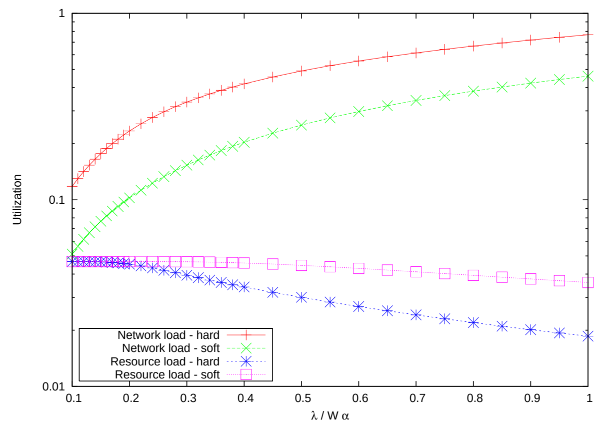

Fig. 7. Network and resource utilization for varying generated network load and fixed mean generated resource load ($\frac{\lambda}{C\beta} = .01$)

can evidently be replaced by other distributions if sufficient evidence can be gathered. Based on the blocking probabilities of individual network links and resources, we can obtain an estimate for the global job blocking probability (B[i]). This process is repeated until two successive iterations achieve an estimate for the global blocking which are sufficiently close to each other. The accuracy of the approximation can, as such, be varied by determining an appropriate value of this parameter . This technique is generally referred to as fixed point approximation. For more details on the convergence of the fixed point technique in this modelling approach, the reader is referred to [14].

The basic European topology, depicted in Figure 5, was used for our validation. This network is composed of 28 network routers and 41 bidirectional links. Each router has a client attached with a fixed job arrival rate ($\lambda_s = \lambda = 1000$ jobs per second). Six resources are installed at a fixed location (routers: Amsterdam, Paris, Berlin, Budapest, Rome and Madrid), and have a fixed processing rate (i.e. $\beta_r = \beta$) depending on the load scenario. Each resource r contains $C_r = 20$ CPUs, while each network link l has $W_l = 20$ wavelengths and a fixed transmission rate ($\alpha_l = \alpha$), also depending on the specific load scenario. As mentioned previously, we implemented a uniform scheduling policy, i.e. $d_{sr} = \frac{1}{|R|}$, and shortest path routing was used for all results.

Figure 6 shows the job blocking probability for varying generated network loads and a fixed mean generated resource load ($\frac{\lambda}{C\beta} = .01$ which implies resource blocking should be negligable). This varying load $\frac{\lambda}{W\alpha}$ can also be interpreted as a varying link dimensioning, i.e. $\frac{\lambda}{W\alpha} \in [0,1]$ is equivalent to $\alpha \in [\frac{W}{\lambda}, \infty]$ for fixed values of W and λ. An immediate observation is the accuracy of the proposed model in comparison to the simulation results. Another important conclusion is that the soft destination approach clearly outperforms hard destination assignment. The difference in blocking behaviour can clearly be attributed to network blocking events (see Figure 7). Soft destination assignment makes use of resource capacity as soon as possible, and as such generates a lower utilization of the transport network. In summary, soft destination improves the blocking behaviour whenever network capacity is the limiting factor.

In Figure 8, the job blocking probability is shown for varying generated resource loads and a fixed mean generated network load ($\frac{\lambda}{C\beta} = .01$). Similarly to the previous discussion, we can conlude the validity and accuracy of the reduced load model. The soft destination approach initially shows worse blocking behaviour than hard destination assignment, which is a consequence of the limited availability of resource capacity. Indeed, jobs with intermediate resources on their path toward their soft destination, will almost always be processed on that resource. However, jobs without intermediate resources on their path will arrive at their soft destination which is experiencing an increased utilization, and thus a higher job blocking probability. This is confirmed by Figure 9, which shows the increased resource utilization for soft destination

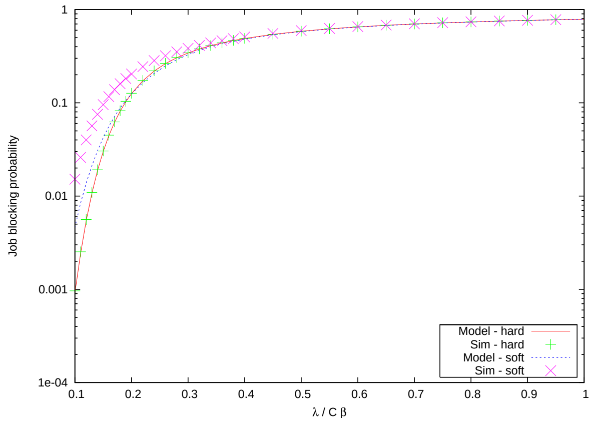

Fig. 8. Job blocking probability for varying generated resource load and fixed mean generated network load ($\frac{\lambda}{W\alpha} = .01$)

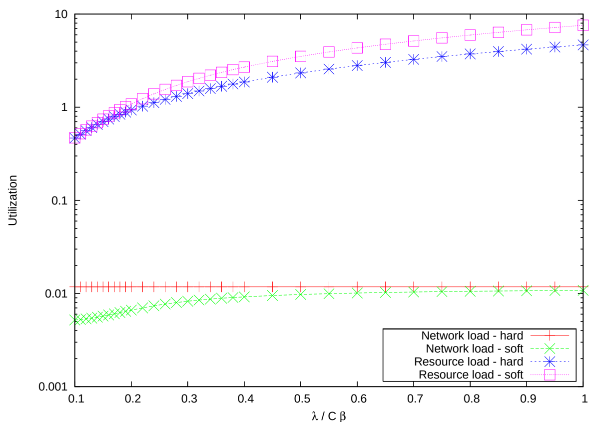

Fig. 9. Network and resource utilization for varying generated resource load and fixed mean generated network load ($\frac{\lambda}{W\alpha} = .01$)

assignment, although network utilization is decreased. This shortcoming of the soft destination approach is likely to be resolved by incorporating algorithms for advanced routing and intelligent resource dimensioning.

III. Dimensioning Algorithms

This section details different optimization techniques for the design of optical Grid networks. The first, using concepts from Divisable Load Theory, allows scalable dimensioning of well-defined OCS-based network scenarios. Afterwards, we present a more general model for combined scheduling and dimensioning.

A. Divisable Load Theory

Given a Grid network architecture, the question arises how operators should decide on the capacities of the network, computational and storage resources at each site: given a number of sites, each characterized by a given job arrival pattern, how much storage/processing capacity needs to be installed at each site, and how should the interconnecting network be dimensioned? The solution should minimize the cost of the entire network. The complexity of solving the Grid dimensioning problem to a large extent stems from the high degree of freedom caused by the anycast routing paradigm. Indeed, given the total aggregated job load, it is possible to calculate the total required processing power, but not where to place it. Work in progress will compare various choices in locating the server capacity, and its impact on network cost. A sample of a dimensioning problem we have already tackled is that of dimensioning for single site excess load [15]. In that case study, each local Grid site was sized to accommodate for a given steady-stat load of locally arriving jobs. Then, for each site in turn it is assumed that that particular site suffers from excessive locally generated load. This excess load is assumed to be distributed evenly among k of the other sites. The resulting network interconnecting all the sites is calculated as the minimal cost solution covering all excess load cases. Three methods for dimensioning were considered: an Integer Linear Programming (ILP) model, a Heuristic and a Divisable Load Theory (DLT) based approximation. While the heuristic is very simple, an equal reduction in complexity is achieved by DLT, but with costs much closer to the true optimal solution ILP. For details, we refer to [15].

B. Combined Scheduling and Dimensioning

Whereas the previous optimization technique focused on solving a specific network scenario, the following presents the solution to a more general problem [16]. The problem we will solve is the following:

a) Given:

- A graph representing the network topology (nodes representing Grid sites and switches, links the optical fibers interconnecting them),
- The arrival process of jobs originating at each site,
- The job processing capacity of a single server, and
- A target maximum job loss rate

b) Find:

- The amount of Grid servers at each site, and
- The amount of link bandwidth to install,
- While meeting the maximum job loss rate criterion.

We take an iterative dimensioning approach, first calculating the amount of server sites needed, and subsequently deriving the inter-site job rates, hence bandwidth. Backed by real world Grid measurements, we will assume Poisson job arrivals [17].

Here, we do not take into account buffering: if at job arrival no free server is found, the job is lost. Thus, assuming Poisson arrivals (mean arrival rate λ), and exponentially distributed job processing times, we use the Erlang-B formula to calculate the total number of servers n required to achieve a maximal loss rate L. To place the n servers among the N sites, we consider three strategies:

1) unif: uniformly distribute the servers among all Grid sites (put $\frac{n}{N}$ at each site);
2) prop: distribute the servers proportionally to the arrival rate at each site (if λ_s is the job arrival rate at site s, then put $\frac{n \cdot \lambda_s}{\lambda}$ servers at site s);
3) lloss: try and achieve the same local loss rate at each site, i.e. use Erlang-B to calculate n_s as the number of

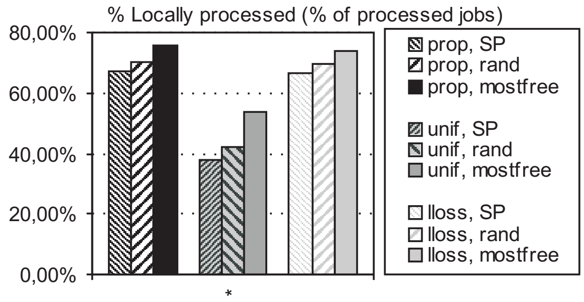

Fig. 10. Fraction of jobs that are processed locally (i.e. at originating site)

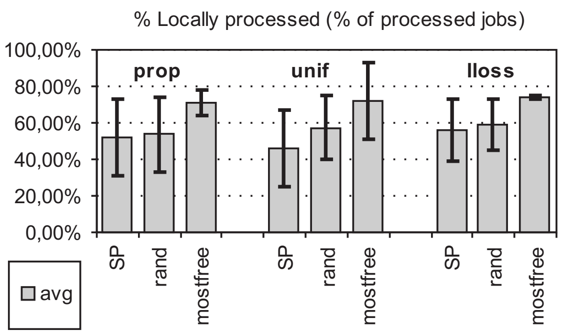

Fig. 11. Local processing fraction averaged over all sites, error bars indicate stdev

servers to install locally at site s to achieve loss rate L, and install $\frac{n \cdot n_s}{\sum n_s}$ servers.

The scheduling algorithm decides where a job is executed. All scheduling approaches studied here will always choose a local server (i.e. at the job arrival site) if one is free. The approaches only differ in electing a remote server for job execution:

1) rand: randomly choose a free server (i.e. among K free servers, each has $\frac{1}{K}$ chance);
2) SP: the closest free server in terms of hop count is chosen, thus striving to minimize network usage;
3) mostfree choose a free server at site S, where S is the site with the highest number of free servers, in an attempt to avoid overloading sites and thus limiting non-local job execution.

We performed a case study on a European network topology with 37 nodes and 57 bidirectional links (this is an extended version of the topology shown in Figure 5). The job arrival rates at each site were chosen randomly (each rate was with 30% chance uniformly chosen in [1,15] and 70% from [30,60]). The first criterion to judge the scheduling and dimensioning strategies by is the amount of jobs, taken over all sites, that is processed locally, shown in Fig 10. Note the relatively low fraction of locally processed jobs, due to the absence of buffering and the high resource load (scaling the arrival rates down to 90%, we achieve 70% local processing).

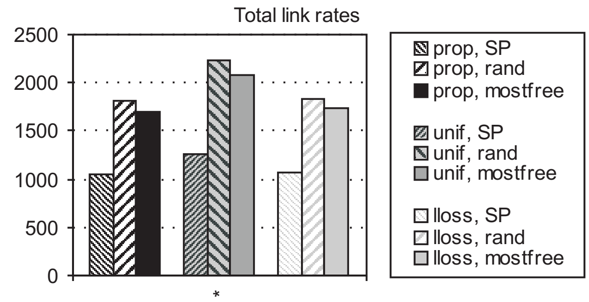

Fig. 12. Total link rates, i.e. number of jobs per time unit crossing each link summed over all links

As intuitively expected, the prop and lloss strategies (placing more servers at sites where more jobs originate) achieve higher local processing rates. From the variation on local processing rates over all sites (see Figure 11), we learn that lloss achieves its aim of equalizing local processing rates, esp. for the mostfree scheduling strategy. From the scheduling perspective, mostfree confirms our intuition by achieving the highest local processing rates. Still, the difference with the others is rather limited. SP, by its deterministic order in choosing sites for remote processing, systematically (over)loads the same servers, thus achieving the lowest local rates.

The last step in the dimensioning process is determining link bandwidths. Using the site-to-site job rates, either an OBS or OCS network can be appropriately dimensioned using conventional methods, e.g. using the Erlang-B formula to calculate the number of wavelengths on each link. (In this particular study using shortest path routing, the amount of wave-lengths for OCS is a factor 5 higher.) In Figure 12 we present the total amount of jobs crossing each link. As expected, the SP scheduling achieves the lowest network load, by minimizing the path length that jobs have to cross. Mostfree obviously achieves lower network loads than rand due to its higher local processing rates, but by ignoring the network topology never comes close to SP. Note the striking impact of choosing an appropriate scheduling strategy: relative differences are bigger than comparing different dimensioning approaches.

IV. Anycast Routing

A. Destination Assignment

The notion of anycast routing amounts to the following: a client submits a job to an anycast address and the (e.g. OBS) network is responsible to provide delivery to at least one, and preferably one, of the suitable Grid resources accepting jobs for the cited address. In [7], three distinct burst destination assignments are compared: soft (SA), hard (HA) and no assignment (NA)[3]. In SA, the source selects a destination, but this can be altered by other nodes along the route to avoid contention or starvation. In HA, this change is not

[3] Soft and hard destination assignment is similar to the concepts presented in Section II-B

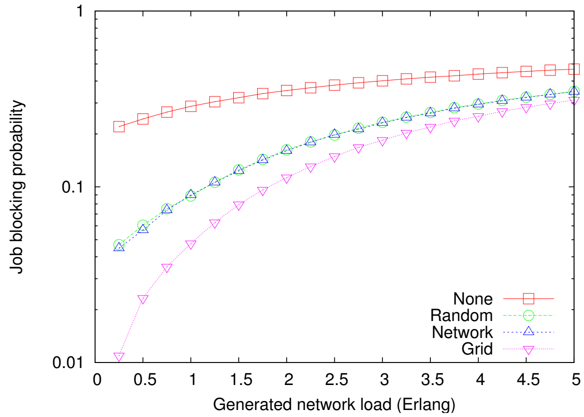

Fig. 13. Comparison of deflection strategies

allowed, and NA allows each node to process the job. Also for contention/starvation resolution, multiple approaches are compared. The SA approach achieved the lowest job blocking probability. Among the deflection approaches, a weighted Grid resource deflection (WGD) algorithm shows the best performance (Figure 13). The presented technique selects the port which has the most options to reach a nearby free resource, by using the following weight function Γ_p for port p at node i:

$$\Gamma_p = \sum_{j,j \neq i} \frac{\Omega_j}{H_p(i,j)},$$

In the previous expression, Ω_j represents the available Grid resources at node j and $H_p(i,j)$ the shortest path hop count from node i to j.

B. Multiple Constraints Routing

In real world Grids, jobs will impose multiple constraints on a Grid site. Thus, a need arises for a solution to the multiple constraints anycast routing problem. In [18], we show how to reduce the anycast problem to unicast routing and propose an extension of a Self Adaptive Multiple Constraints Routing Algorithm (SAMCRA) with a new non-linear length function guaranteeing exactness. A distributed variant is shown to achieve routing results close to the pseudo-optimal solution obtained with a maximum flow algorithm.

Several routing algorithms are proposed:

- SAMCRA*, an update of the SAMCRA algorithm
- Maximum flow pseudo-optimal bound
- Best Server and Best Delay heuristics

SAMCRA or Self-Adaptive Multiple Constraint Routing Algorithm is an online algorithm to determine the shortest path subject to multiple constraints [19]. Unfortunately, its traditional method of ordering subpaths (based on a non-linear length function) can cause sub-optimal results, eventually leading to routing loops [20]. A novel path ordering, which guarantees optimality, is therefore introduced and the resulting algorithm is named SAMCRA*. In its original form, SAMCRA(*) can only be applied to unicast routing problems. Extending the algorithm for anycast routing requires the introduction of a virtual topology, consisting of a virtual resource linked to all physical resources. Each client will then route towards that virtual destination. Finally, note that SAMCRA(*) is available as a source-based, centralized algorithm, making routing decisions for the whole network on the edge routers, or as a sub-optimal, distributed hop-by-hop version, executed on each participating network router.

Maximum flow, due to Ford and Fulkerson, is an optimal, offline technique to determine the maximum amount of flows between a given source and destination. It essentially locates paths between source and destination with free capacity (referred to as augmenting paths), and routes as many flows as possible over these paths. Similar to SAMCRA(*), supporting the anycast scenario also requires the incorporation of a virtual resource, whereby the capacity of the virtual links is proportional to the processing rate of the attached resource. In case job characteristics of individual clients (e.g., required processing capacity and average runtime) remain identical, a virtual source can be introduced in the network, together with links connecting the virtual node to the physical clients. Virtual client link capacities are proportional to the job arrival rate of the attached client, while virtual resource link capacities are proportional to the job processing rate of the attached resource. In effect, this allows the use of the (classical) single-commodity, maximum flow algorithm. However, in case job characteristics differ between clients, a virtual client cannot be introduced and a multi-commodity, maximum flow algorithm needs to be used between all clients and the single, virtual destination. In the following, we only consider the single-commodity, maximum flow algorithm. Finally, the introduction of a deadline as job constraint causes the pseudo-optimal behaviour of the maximum flow technique. Indeed, paths violating the deadline constraint are not considered as a possible augmenting flow path, and thus the true maximum flow is not attainable.

Heuristic techniques, implementing straightforward strategies for resource and path selection, are introduced for comparison purposes. First, in Best Server, the client selects the server with the highest available capacity, and uses fixed shortest path routing to reach that server. In contrast, the client selects the server that can be reached within the smallest network delay in the Best Delay approach.

As shown on Figure 14, the acceptance rate of the intuitive heuristics Best Server and Best Delay is much lower than both SAMCRA* variants. When wavelengths are sparse, Best Delay can approach SAMCRA*'s acceptance probability. Unfortunately, as network capacity increases, job requests are frequently scheduled on overloaded resources. The Best Server heuristic consumes too much network resources, and therefore converges only slowly to a maximum acceptance probability for an overdimensioned network. The close match between the SAMCRA* scheduling results and the maximum flow pseudo-optimal bound emphasizes the effectiveness of this algorithm. Figure 15 illustrates that SAMCRA* steers a middle course from the path delay perspective, while still satisfying the end-

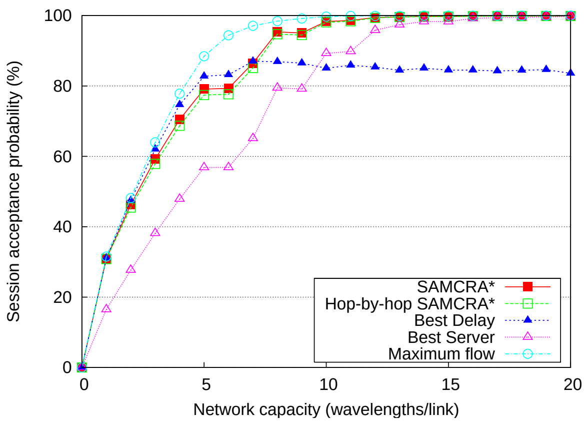

Fig. 14. Session acceptance probability vs network capacity for different anycast algorithms

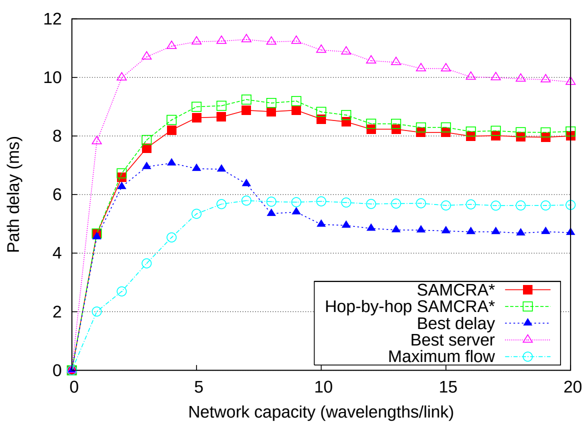

Fig. 15. Average session delay vs network capacity for different anycast algorithms

to-end delay requirements. For detailed information on the simulation parameters, the reader is referred to [18].

V. Conclusion

This paper addressed several fundamental questions related to the optimization of emerging optical Grid networks. We presented the architecture and the different network layers, and briefly discussed the effects on the optical switch architecture. We presented a performance model to evaluate the blocking probability of optical Grids, and we showed the accuracy and scalability of the model through comparison with simulation analysis. Two major goals were achieved in our work on dimensioning algorithms. First, a scalable modeling technique (based on Divisible Load Theory) was shown to converge towards the optimal ILP-formulated model. Additionally, we established the importance to combine dimensioning and scheduling algorithms in the design phase of the network. Finally, the relevance of anycast routing was demonstrated, by showing a practical algorithm (both in centralized and distributed form) which is able to achieve near-optimal job scheduling over optical networks.

Acknowledgment

This work was supported by the Commission of the European Union through the IST-FP6 Network of Excellence e-Photon/ONe. M. De Leenheer thanks the IWT for their financial support through his Ph.D. scholarship. C. Develder thanks the FWO for his post-doctoral grant.

References

[1] C. Qiao, and M. Yoo, *Optical Burst Switching: A New Paradigm for an Optical Internet*, Journal of High Speed Networks, 8(1):69-84, Mar 1999.

[2] R. Nejabati (editor), *Grid Optical Burst Switched Networks* Grid High-Performance Netwroking Research Group (GHPN-RG), Open Grid Forum, Jan 2007.

[3] *http://www.ist-phosphorus.eu*, Website

[4] D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, A. Tzanakaki, M.J. O'Mahony, *Dynamic Optical Network Architectures and Technologies for Existing and Emerging Grid Services*, IEEE Journal of Lightwave Technology, 23(10):3347-3357, Oct 2005.

[5] N. Barakat, and E.H. Sargent, *Separating Resource Reservations from Service Requests to Improve the Performance of Optical Burst-Switching Networks*, IEEE Journal on Selected Areas in Communications, 24(4):95-107, Apr 2006.

[6] F. Xue, S.J.B. Yoo, H. Yokoyama, and Y. Hoiuchi, *Performance Comparison of Optical Burst and Circuit Switched Networks*, Proc. of the Optical Fiber Communication Conference (OFC), Anaheim, CA, USA, Mar 2005.

[7] F. Farahmand, M. De Leenheer, P. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, J.P. Jue *A Multi-Layered Approach to Optical Burst-Switched Based Grids* Proc. of the Fifth International Workshop on Optical Burst/Packet Switching (WOBS), Boston, MA, USA, Oct 2005.

[8] J. Baert, M. De Leenheer, B. Volckaert, T. Wauters, P. Thysebaert, F. De Turck, B. Dhoedt, P. Demeester, *Hybrid Optical Switching for Data-Intensive Media Grid Applications*, Proc. of the COST291/GBOU ONNA Workshop on Design of Next Generation Optical Networks: from the Physical up to the Network Level Perspective, Gent, Belgium, Feb 2006

[9] E. Van Breusegem, J. Cheyns, D. De Winter, D. Colle, M. Pickavet, F. De Turck, P. Demeester, *Overspill routing in optical networks: a true hybrid optical network design*, IEEE Journal on Selected Areas in Communications, 24(4): 13-25, 2006.

[10] M. De Leenheer, P. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, and M.J. O'Mahony, *A View on Enabling Consumer Oriented Grids through Optical Burst Switching*, IEEE Communications Magazine, 44(3):124-131, Mar 2006.

[11] D. Simeonidou, G. Zervas, and R. Nejabati, *Design considerations for photonic routers supporting application-driven bandwidth reservations at sub-wavelength granularity* Proc. of the Workshop on Optical Burst Switching (WOBS), San Jose, USA, Oct 2006.

[12] M. De Leenheer, C. Develder, F. De Turck, B. Dhoedt, P. Demeester, *Erlang Reduced Load Model for Optical Burst Switched Grids*, Accepted for publication in Proc. of the First Int. Workshop on GRID over Optical Burst Switching Networks (GOBS), Athens, Greece, Jun 2007.

[13] C. Partridge, T. Mendez, W. Milliken, *Host Anycasting Service*, RFC 1546, Internet Engineering Task Force.

[14] Z. Rosberg, H.L. Vu, M. Zukerman, and J. White, *Blocking Probabilities of Optical Burst Switching Networks Based on Reduced Load Fixed Point Approximations*, Proc. of IEEE Infocom, Vol. 3, pp. 2008-2018, Mar 2003.

[15] P. Thysebaert, M. De Leenheer, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, *Scalable Dimensioning of Optical Transport Networks for Grid Excess Load Handling*, Photonic Network Communications, 12(2):117-132, Sep 2006.

[16] C. Develder, M. De Leenheer, T. Stevens, B. Dhoedt, F. De Turck, P. Demeester, *Scheduling in Optical Grids: A Dimensioning Point of View*, Accepted for publication in Proc. of the Conference on the Optical Internet - Australian Conference on Optical Fibre Technology (COIN-ACOFT), Melbourne, Australia, June 2007

[17] K. Christodoulopoulos, M. Varvarigos, C. Develder, M. De Leenheer, and B. Dhoedt, *Job Demand Models for Optical Grid Research*, Accepted for publication in Proc. of the 11th Conference on Optical Network Design and Modelling (ONDM), Athens, Greece, May 2007.

[18] T. Stevens, M. De Leenheer, F. De Turck, B. Dhoedt, and P. Demeester, *Anycast Routing Algorithms for Effective Job Scheduling in Optical Grids*, Proc. of the European Conference on Optical Communication (ECOC), Cannes, France, Sep 2006.

[19] P. Van Mieghem and F. Kuipers, *Concepts of Exact Qos Routing Algorithms*, IEEE/ACM Transactions on Networking, 12(5):851-864, Oct 2004.

[20] T. Stevens, M. De Leenheer, F. De Turck, B. Dhoedt, and P. Demeester, *Distributed Job Scheduling based on Multiple Constraints Anycast Routing*, Proc. of the Third International Conference on Broadband Communications, Networks, and Systems (Broadnets), San Jose, CA, USA, Oct 2006