034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable reference number: D.5.7

Grid Network Design

Due date of deliverable: 2008-09-30
Actual submission date: 2008-09-30
Document code:<Phosphorus-WP5-D.5.7>

Start date of project: October 1, 2006	Duration: 30 Months

Organisation name of lead contractor for this deliverable: **Athens Information Technology (AIT)**

Revision [v1.0]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract

This deliverable focuses on the design of long reach, efficient and high computational capacity lambda Grids. More precisely, two fundamental design areas are investigated: the optimization of the location and capacity of computational/network resources (dimensioning) and the architectural aspect of selecting an efficient switching scheme. In this context, novel approaches for solving various practical formulations of the Grid dimensioning problem are proposed and evaluated through simulations. The cost and efficacy of a lambda Grid with regard to the switching paradigm employed is investigated through studying – both qualitatively and quantitatively - the deployment of specific Phosphorus applications. Beyond homogeneity in terms of switching schemes used, the adoption of hybrid solutions is shown to broaden the design space, enabling trade-offs of finer-granularity between performance and network cost. Additionally, a novel framework for comparing fundamental switching techniques is presented; the latter is used to rate the efficiency of circuit- vs. burst-switching in the context of lambda Grids.

List of Contributors

Kostas Katrinis	AIT	Balagangadhar Bathula	ULeeds
Anna Tzanakaki	AIT	Jaafar Elmirghani	ULeeds
Jens Buysse	IBBT	Manos Varvarigos	RACTI
Chris Develder	IBBT	Panagiotis Kokkinos	RACTI
Marc De Leenheer	IBBT	Kostas Christodoulopoulos	RACTI
Taisir El Gorashi	Uleeds	Michael Kalochristianakis	RACTI

Table of Contents

List of Figures

List of Tables

0 Executive Summary

This deliverable documents the outcome of collaborative research endeavours towards the design of the optical network infrastructure supporting an efficient, large-scale and of high-capacity Grid. In this context, this work extends the current state-of-the-art with regard to dimensioning of Grid computational and network resources that is required to offer adequate service quality in a cost-efficient manner.

Part I is devoted to the dimensioning of computational and network resources in the scope of Phosphorus. Although the problem of dimensioning a transport network is not new, the consideration of signal quality degradation caused by the optical medium calls for revisiting the problem. A novel method for impairment-aware dimensioning of WDM networks is presented for the first time, which is shown to reduce the total capital cost of the Grid network. Furthermore, certain characteristics of Grid applications such as anycast routing, scheduling policies and strict execution times, in combination with the wide range of Grid-related application, allow for improved location and dimensioning of computational/storage resources. Since the placement and configuration of computational resources forms part of the input to the problem of dimensioning the Grid network, a concise study of the problem proves more than necessary. New algorithms for efficiently locating and assigning capacity to computational resources are presented. While there is high value in studying the two dimensioning sub-problems (computational vs. network resource dimensioning) separately, applying the related methods sequentially does not guarantee optimality for the unified problem. To overcome this, a novel method that jointly dimensions Grid computational and network resources is presented and evaluated through simulations.

Part II shifts the Grid network design focus to architectural aspects, namely the selection and configuration of optical switching paradigms. A thorough survey of available switching techniques (circuit vs. burst switching) is presented, together with new algorithms specific to the burst switching primitive. The suitability of each of these techniques with regard to the requirements of particular Phosphorus reference applications is qualitatively rated through case studies. Also, quantitative results with regard to special instances of the network design problem – such as traffic heterogeneity and QoS in anycast routing – are presented. Departing from the use of a single switching technique throughout an optical Grid network, the cost vs. performance trade-off of using combinations of different types of switching elements within the same network is explored, indicating that the use of a hybrid approach can bring substantial savings to the total network cost.

Part I

Similar to any engineering problem, the process of designing a networked Grid entails various cycles of performance vs. cost decisions. Among these, the optimization of the capacity put on deployed machinery - also known as dimensioning – is critical towards fulfilling design requirements, while controlling incurred cost. Although the problem is not new, specific idiosyncrasies of an optical Grid necessitate revisiting the problem in the context of Phosphorus: new networking constraints are created by the optical infrastructure, while high-speed connections create new constraints for distributed computation.

The problem of Grid dimensioning can be logically viewed as a two-stage process: dimension first computational/storage resources and then optimize capacity of network resources. Solving these two sub-problems jointly yields optimal results. On the other hand, solving these two sub-problems sequentially can be practical in case either of the two dimensioning problems is already solved. In any case, novel approaches that solve one of the two sub-problems separately are valuable in the sense that they can be used as building blocks towards more efficient unified dimensioning methods.

Section 1 formulates the two sub-problems of Grid dimensioning, namely (computational) resource and network dimensioning. In addition, the problem of solving the two sub-problems jointly is positioned. Section 2 proposes novel approaches for solving the two dimensioning sub-problems, both jointly and in separation.

1 Introduction into Grid Dimensioning

1.1 Resource Dimensioning

1.1.1 Dimensioning Issues

In the context of Resource Dimensioning we are interested in the dimensioning issues related to the computational and the storage resources of a Grid Network. A number of different dimensioning issues can be considered:

1. The in-advance knowledge of the number of computational and storage resources needed to satisfy users requirements is helpful for planning/expanding the Grid network and is a key issue when offering new services. An important goal is the maximization of the resource utilization, which leads to the minimization of the number of computation and storage resources needed for task execution. Similarly challenges arise in cloud computing and storage environments, where execution and storage spaces can be created on the fly by partitioning resources (e.g., computational capacity) and resource containers.

2. The proper placement in the network of the computational and storage resources is another dimensioning issue. For example, data intensive applications the proper placement in the network of the computational and storage resources is another dimensioning issue. For example, such an issue may arise for data intensive applications. The proper placement of the storage resources in the network (and of course their capacity) influences the applications execution time. A similar problem is the efficient placement of datasets and their replicas in the storage resources of the Grid network. This is very important for applications requiring more than one piece of data and for data redundancy in case of failures.

3. The type of computational and storage resources used is also a very important issue. These kinds of resource can be categorized in various ways. For example by using Flash-based storage devices users can immediately increase application performance and save on energy costs compared to traditional

Fibre Channel storage systems. Furthermore, computation resources can be categorized based on the type of users they serve or the priority they give to each type of users.

4. Dimensioning also relates to the effect of various computation or storage related parameters on the performance of the Grid Network and the related algorithms.

1.1.2 Related Work

A number of works exist investigating the resource dimensioning issues introduced in Section 1.1. Issue 1 is usually investigated in the form of scheduling algorithms trying to achieve efficient use of the computational or storage resources, by minimizing the number of resources needed to serve all tasks. These approaches try to maximize overall system performance (that is, Grid resource utilization), without taking into account users and tasks Quality of Service (QoS) requirements. Advance reservation of resources, which is the ability of the scheduler to guarantee the availability of resources at a particular time in the future, is one mechanism Grid providers may employ in order to offer specific QoS guarantees to the users [Buyya02] [Ali04] [Varvarigos05]. However, these algorithms lack scalability, as they are unable to efficiently perform task scheduling in short time for large numbers of Grid resources. Using concepts from computational geometry, [Rouskas07] addresses the scalability problem for task scheduling under a user's satisfaction framework. The scalability problem is also addressed in [Zhang06].

Issue 2 is investigated in Grids mainly in the form of data placement and data management in general. Data management in Grids deals mainly with data migration, that is, with the decisions regarding the place and the time application-related datasets should be moved. The effects of data migration in Grids have been considered in [Biswas04]. The most common data migration technique is data replication [Zini02], which is the process of distributing replicas of data across sites. When different sites hold replicas of a particular dataset, significant improvements can be realized by selecting the best replica among them. The best replica is the one that optimizes a desired performance criterion such as access latency, cost or security [Zini02][Barker07] [Foster02].

The usage of different kinds of computational resources (Issue 3) has been investigated in [D.5.2] and [Kokkinos07]. Specifically, we propose a Quality of Service (QoS) framework for Grids, where computational resources are distinguished based on the type of users they serve and the priority they give to each type. The proposed framework provides hard delay guarantees to Guaranteed Service (GS) users and user fairness to Best Effort (BE) users.

Finally, Issue 4 relates to the effects, caused by changes in various computational or storage related parameters, on the performance of the Grid network. This issue is examined more or less (directly or indirectly) by all the Grid related algorithms.

These resource dimensioning issues are investigated in more detail in Section 2.2.2.

1.2 Network Dimensioning

1.2.1 Background and Motivation of Work

The process of designing a networked Grid entails inevitably solving a network dimensioning problem: specify the capacity that needs to be installed in the network to be able to seamlessly carry the estimated volume of traffic generated by the Grid sites. Specific to the lambda-switched Phosphorus Grid, given an input physical topology and an amount of traffic requests in the form of a traffic matrix, following network resources are to be dimensioned:

- Number of fibres that need to be installed/activated among any pair of optical switches in the physical topology.

- Number of wavelengths that need to be installed on each installed/activated fibber.

- Number of input/output ports of the switch fabric for each optical switch in the physical topology.

Depending on whether the optical infrastructure is pre-installed or not, two distinct formulations of the Grid network dimensioning problem are possible: a) the “deployed network” problem, where already installed fiber is assumed at each physical link of the input topology and b) the “greenfield” problem, where all network-related infrastructure will be installed from scratch. In the second scenario, the set of physical links in the input topology constitutes only a set of candidate links among switching nodes and the dimensioning process has to additionally specify the optimal set of physical links.

The problem of dimensioning a Grid network can be viewed as an optimization problem that seeks to specify the capacity assigned to the previously listed resource types (and a set of physical links wherever this applies) in a manner, such that the total capital and operational cost of the Grid network is minimized, while all traffic demands are deterministically served. While such a formulation follows naturally from the fundamentals of network optimization theory, certain particularities of the optical medium exacerbate the problem: physical impairments may degrade the quality of the signal carried by the optical network, leading to either excess overhead to higher layers or worst-case to practically unusable paths. In any case, the effect of physical impairments could be detrimental to a Grid application. In section 2 it is shown that the extent of impaired paths can be extremely high for particular configurations of the Phosphorus testbed.

The above pathogeny motivates the incorporation of impairment-awareness to the problem of network dimensioning. The use of selective optoelectronic regeneration is a common approach towards cancelling the effect of optical impairments. Still, two critical issues related to regeneration remain unresolved in existing state-of-the-art, namely:

1. Specifying the optimal point in the network design process, where placement of regenerators should occur. While it is simpler to apply regeneration to an already dimensioned network (post-dimensioning

regeneration), it is not obvious that this decision is the most efficient in terms of total cost, compared to **carrying out dimensioning and regenerator placement jointly**.

2. Devising a network dimensioning method that guarantees acceptable signal quality and **at the same time being tolerant to the heterogeneity of the optical components** deployed throughout an optical network. As it will be shown later, existing approaches based on optical reach as a criterion for regeneration fail to satisfy both of these two requirements.

Section 2.2.3 elaborates in these issues in the context of Phosphorus, proposes solutions and evaluates their benefit through experimentation.

1.2.2 Related Work

Literature on dimensioning the network infrastructure in support of an optical Grid is scarce. The authors of [Thysebaert05] test the applicability of established optimization methods in the context of Grid network dimensioning and propose using Divisable Load Theory towards designing tractable algorithms, showing the incurred benefit through simulation. Otherwise, past work on regenerator placement and design of translucent optical networks [RamaFe99] is closely related to the problem of dimensioning a Grid network.

Many of the aspects and trade-offs for designing core optical networks optimally are addressed in [Simmons06], where among others general guidelines for selective regeneration are given. The work in [VanPar01] demonstrates clearly the trade-off between mass of regeneration and lightpath capacity in real networks although it is mostly related to routing rather than network design, [Yetgin03] presents ILP and heuristic solutions for both network design and connection provisioning sub-problems. [Filho03] addresses the problem of minimum regeneration cost network design in two deployment scenarios (all nodes candidates for regeneration vs. “transparency islands”) and [Yang05],[Shen02] provide approximate solutions to related problems. The problem of designing translucent networks with transparency islands owned by more than a single carrier is dealt with in [Chat07] through a game-theoretic approach. Last, [Carp03] and [Savasi07] approach the problem of regenerator placement as a connected dominating set instance and show reduction in total regeneration cost. All the above work has contributed considerably to modelling optical networks with selective regeneration and has provided exact and approximate solutions to their design. However, it is based on the simplifying assumption that the physical distance covered by a lightpath is a good approximation of signal quality degradation. While this is realistic for limited hop-count lightpaths and for networks exhibiting high homogeneity in terms of deployed optical equipment (e.g. amplifiers, de-/multiplexers) and their configuration, there are still various realistic scenarios, where the approach of designing translucent optical networks using a single optical reach threshold as regeneration criterion is not valid.

Zang et al. [Zang02] present various heuristics for sparse regeneration, using an analytical model for amplified spontaneous emission noise to specify the set of switching nodes, where regeneration should occur. [Birkan06] elaborates in minimum cost signal rectification along fiber links due to Polarization Mode Dispersion (PMD) degradation. Although focusing primarily on regeneration, both of these studies point to the direction of impairment-aware network design. The motivation and results found in [Ali02] and [Morea04] are the most closely and explicitly connected to the contribution of the present work in terms of network dimensioning. More precisely, [Morea04] tests the correlation between optical reach and network performance, using analytical modelling of impairments. Although not taking a minimum cost approach concerning network design, still the aforementioned work has great value in modelling and evaluating the impact of a group of physical impairments to optical network design. Last, the inquiry of the impact of PMD to network design in [Ali02], while not having the same objectives with the work presented in this document, is of great significance in the scope of impairment-aware network design for two reasons: a) it presents an explicit model and analytical formulation of a network design problem that takes impairments into consideration and b) it indicates – albeit implicitly – that optical equipment heterogeneity is an important aspect of network design that is not to be neglected.

1.3 Holistic (Joint) Grid Dimensioning

1.3.1 Motivation and Challenges

For all the volume of work that has been conducted on network dimensioning, the inter-relation between application behavior/topology and network dimensioning has not received much attention. This is particularly the case in Grid network dimensioning, where the location and capacity of Grid sites affects the input traffic matrix assumed during solving the network dimensioning problem. To fill this gap, joint dimensioning of computational and network resources is studied in section 2.2.1.

1.3.2 Related Work

Work on dimensioning Grids is scarce. In [Thysebaert05] analytical ILP and heuristic approximations are used to cater for excess load: it is assumed that each of the Grid sites (dimensioned for the locally generated jobs) may suffer from overload, and network dimensions (number of wavelengths and fibres used) are determined by finding a global optimum over all single-site overload problems.

One way to deal with the unknown destination for Grid jobs is to assume that the fraction of jobs (originating at a particular site) going to a given computational Grid site is known, thus fixing a priori the arrival rates of jobs at each job execution site. This approach is taken in [DeLeenheer07], where an analytical methodology known as reduced load fixed-point approximation [Rosberg03] is used to dimension both network and computational resources.

Existing literature (to the best of our knowledge) however does not discuss a 'clean slate' or greenfield Grid dimensioning problem, finding the complete Grid capacity required to meet a given Grid job arrival pattern. Also, we did not find any results assuming fully flexible scheduling strategies without any knowledge of probabilities for selecting a given destination site. Our own contribution (published in [Develder08]) is discussed in detail in Section 2.2.1**Błąd! Nie można odnaleźć źródła odwołania.**.

2 Grid Dimensioning – The Phosphorus Approach

2.1 Requirements Specification

2.1.1 Dimensioning Considerations

A number of important issues should be considered when dimensioning lambda Grids. These issues are related to both computational and network resources.

The specificities of the optical medium in support of a Grid call for considering quality degradation of the signal on the optical medium. Physical impairments such as amplifier noise, crosstalk, dispersion, filter concatenation effects and other non-linear impairments have to be taken into account. What is more, the effect of such impairments is magnified as the geographical reach of the network increases. Given the wide reach of Grid networks, considering physical impairments as a parameter during network dimensioning becomes inevitable. Impairment-aware network dimensioning is considered in Section 2.2.3.

An important aspect of Grid dimensioning is that in addition to the network dimensioning, the computational and/or storage resources need to be dimensioned. The number of resources installed and their locations should be optimized. The location of resources determines the traffic matrix as they impact, where jobs will be processed. Resource dimensioning is considered is Section 2.2.2 where a scheduling algorithm

As it will be made clear in this section, the allocation of network capacity directly depends on the expected traffic matrix, which in turn depends on resource dimensioning. Clearly, there is an inter-relation between the two optimization problems, calling for solving them jointly. Joint dimensioning is considered in Section 2.2.1.

Resilience constraints protecting against possible failures is another important issue to be considered when dimensioning a Grid. Failure of computational resources, network links and/or optical cross-connects should be taken care of during dimensioning.

Furthermore, It is important to efficiently utilize network resources so that fewer upgrades are required, when traffic changes. Long-term planning, where the Grid evolution over time in response to changes in traffic demand is determined, is an important issue. In long term planning the network topology and capacity expansion is determined over a long time interval spanning multiple years. In [Pickavet99], two different solution methods are presented: a sequential single-period approach, which designs the networks for every time period separately in chronological order, and an integrated multi-period approach, which considers all time periods at once.

Dimensioning algorithms also vary depending on the network technologies and topologies. For example, single- or multi-layer scenarios involving one or more network layers can be considered, such as when jointly dimensioning IP routers and WDM cross-connects [Holler06]. Also dimensioning algorithms with or without grooming [Zhu03] where e.g. IP flows between different end points can share the same WDM circuits over multiple hops, bypassing some of the IP routers.

Different topologies such as ring [Gerstel00] or mesh networks, and different design criteria (e.g. survivability [Colle02], availability) can also be considered.

Last, dimensioning algorithms in single-domain or hierarchical networks are also taken onto account. [Bley04] provides algorithms for deciding how to partition the network in access and backbone nodes, as well as designing the backbone topology.

2.1.2 Problem Statement

Enabling a Grid application depends partly on the type of the application. In [D.2.1] applications are classified according to various criteria. Generally, Grid applications can be classified according to parallelism, e.g. Single Program-Single Data applications represent sequential programs that take a single input set and generate a single output set. Classification can also be based on the frequency and size of communication required for initial movement of data and programs, both prior and during computation. Further classification can be based on the granularity (how long a program can execute before it needs to communicate with other programs in the application) and dependency between programs within an application. The above classifications create a set of requirements with regard to the service, transport, control and management planes [D.2.1]. These requirements reflect the phosphorous vision. The main innovation introduced by Phosphorus is the provisioning of the network (lightpath) and Grid (computational, storage) resources in a single-step. Phosphorus also aims at enabling seamless end-to-end dynamic service provisioning across heterogeneous network infrastructures, by introducing developments in all planes, both in terms of internal functionalities and of exposed interfaces. The above requirements should be considered when addressing dimensioning in the Phosphorus architecture.

In addition, [D.2.1] defines two control plane models for the Grid-enabled GMPLS (G^{2}MPLS) architecture: Overlay and Integrated. These models concern the layering of grid and network resources. While in the Overlay model, most of the computational and service intelligence is maintained on the Grid layer that has Grid and network routing knowledge in order to provide Grid and network resource configuration and monitoring, the integrated model moves most of the functionalities for resource advance reservation to the G^{2}MPLS network control plane. These two different architectures are further aspects to be considered during the process of dimensioning.

2.2 Dimensioning Approaches in Phosphorus

One of the design problems faced when deciding to deploy a Grid is network dimensioning: estimating how much capacity is needed for the network to be able to transport the required level of Grid traffic. Another aspect in Grid dimensioning is that not only the network resources, but also the computational and/or storage resources need to be dimensioned: how many servers and of what capability need to be installed, and on which sites? The following sections will discuss these dimensioning approaches in more detail. First, we describe how to couple/integrate both network and resource dimensioning approaches in a Grid context. Subsequently, we present approaches that solve the network and resource dimensioning problems independently.

2.2.1 Joint Dimensioning

If we want to apply any of the traditional network dimensioning approaches (as discussed poniżej, Section 2.2.3) for dimensioning Grids, the problem arises of accurately estimating the traffic matrix. Indeed, given the anycast principle typical of Grids, the destination of the traffic (i.e. Grid jobs) is not given a priori. Another aspect in Grid dimensioning is that not only the network resources, but also the computational and/or storage resources need to be dimensioned: how many servers need to be installed, and at which sites? Note that the latter will have an impact on where jobs will end up being executed, i.e. the eventual traffic matrix, hence the network dimensions. It is clear that jointly determining both server and network dimensions is a very hard problem (note that even single-period dimensioning, where a single traffic matrix is given specifying the average demand between every node pair, may already be NP-hard [Mukherjee96]). Therefore, we will propose a phased approach, dimensioning first the servers and then the network (see further).

This presents a viable dimensioning methodology, and assesses the impact of the scheduling algorithm on Grid network dimensions. Yet, since development of scheduling algorithms as such is not this work's primary concern, we will assume fairly straightforward scheduling strategies, based on a single scheduler possessing global knowledge, thus being able to find a free server for every arriving job based solely on the job's arrival time and duration, and server processing speed and occupancy. For more advanced scheduling algorithms, including e.g. advance reservation concepts and QoS support, we refer to previous Phosphorus deliverables [D.5.4] and [D.5.3]. We believe that adding QoS support or advance reservations is unlikely to affect the qualitative comparison of the different scheduling strategies discussed further.

2.2.1.1 *A phased solution to the Grid dimensioning problem*

We will take an iterative dimensioning approach, starting with an algorithm for choosing appropriate server site locations: not every Grid site will necessarily be a server site. Next we will calculate the amount of servers needed (and distribute them amongst the chosen server site locations). Subsequently the inter-site job rates are determined and hence required bandwidth. In the work presented, we focus on computational Grids, where jobs consist of a single unit of work submitted to the Grid, characterized by a data size, and a computational requirement (e.g. expressed in number of floating point operations, FLOP). An accurate problem statement is the following:

- 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 CPU (an average of μ jobs/s), and
  - A target maximum job loss rate
- Find:
  - The locations of the server sites,
  - The amount of Grid server CPUs at each site, and
  - The amount of link bandwidth to install,
  - While meeting the maximum job loss rate criterion and minimizing network capacity.

Given the complexity of the problem (e.g. the dependence of the network capacities on the choices of server locations and capacities), we opt for a phased solution approach comprising subsequent steps. The first step will be to find K server locations (out of the N Grid sites), while a second step finds the server capacities at each of the K chosen sites. The third step will determine the amount of jobs exchanged between the Grid sites and the server locations. The final fourth step will be to calculate the actual network dimensions, i.e. link bandwidth. Each of these steps is now discussed in detail.

A. Finding the K best server locations

The aim of the first step in solving our Grid dimensioning problem is to figure out which locations are best suited for placing the servers. The cost criterion to measure by will be the total expected link bandwidth. The major difficulty in evaluating that cost for a given choice of K locations, is that the required bandwidth depends also on the amount of server capacity installed at each of the server sites and possibly the Grid scheduling and routing algorithm. Therefore, we make some simplifying assumptions: (i) each Grid site i will send all its jobs to a single destination D_i, and (ii) shortest path routing is used. Hence, given a choice of K locations, a site i will send its jobs to server site j if the routing distance H_{ij} is the minimum over all H_{ik} values for k = 1..K.

Finding the optimal choice of K sites hence is a k-means clustering problem (or rather a k-medoid problem, since cluster centers are actual data points): we are looking for K cluster centers, the centers representing the server sites, and the cluster members the Grid sites sending their jobs to that center (server). A well-known heuristic k-means clustering algorithm [MacQueen67] solving the problem is rephrased in Figure 1 as a k-medoids algorithm for the Grid site location problem. Repeating this algorithm for various randomly chosen initial K server locations leads to solutions close to the exact solution. However, given the simplifying assumptions, a fairly compact ILP formulation of the problem can be devised, as outlined in Figure 2. Given the relatively small number of (binary) variables and equations ($O(N^2)$ with N the number of sites) , the time to solve it for the case studies we considered (comprising a few tens of nodes) was most acceptable (a few seconds at most). The results we present in this work are obtained using the ILP solution method. (Note that for other, prohibitively large problem instances, the k-medoid heuristic can provide acceptable solutions quite fast.)

Given constants: H_{ij} = routing distance (e.g. hop count) from site i to site j (i, j = 1..N)

λ_i = job arrival rate at site i (i = 1..N)

K = the number of clusters and centers (hence server sites) to choose

(1) Choose K initial medoids m_k (k = 1..K).

(2) Form clusters: assign each object (Grid site) to closest centroid:

For each i = 1..N, assign node i to the cluster set C_k with centroid m_k if

$$H_{i,m_k} = \min\{H_{i,m_l}, l = 1..K\}$$

(3) Recalculate the positions of the K medoids within their cluster:

For each k=1..K, let m_k be that node m in set C_k minimizing $\sum_{i \in C_k} \lambda_i \cdot H_{im}$

(4) Repeat steps 2-3 until the medoids m_k no longer change; these cluster centers are the server locations.

Figure 1: K-medoids clustering algorithm for choosing K server locations. (Note: the term 'k-medoids' is used, instead of 'k-means', since the cluster centers are actual data points—in casu site locations—and not freely chosen means.)

Decision variables: T_j = 1 if and only if site j is chosen as a server site location, else 0
S_{ij} = 1 if and only if site j is the target server for traffic from site i, else 0

Given constants: H_{ij} = routing distance (e.g. hop count) from site i to site j (i, j = 1..N)
λ_i = job arrival rate at site i (i = 1..N)
K = the number of server sites to choose

$$\min \sum_i \sum_j \lambda_i \cdot H_{ij} \cdot S_{ij} \quad \text{with} \quad \begin{cases} \sum_j T_j = K & \text{(only K server locations)} \\ \sum_j S_{ij} = 1 \quad \forall i & \text{(simplifying assumption: for each site i, only send jobs to a single server site j; the objective will ensure it is the closest one)} \\ S_{ij} \le T_j \quad \forall i, j & \text{(only send traffic to server sites)} \end{cases}$$

Figure 2: ILP for choosing K server locations.

B. Determining the server capacities

For determining the amount of required server capacity, it is necessary to make some assumptions on the Grid job arrival process. In this work, we focus on computational Grid jobs and thus we will dimension the servers in terms of processing capacity, expressed in number of CPUs. A job is assumed to fully occupy a single CPU for its entire duration. Furthermore, we will assume that Grid job requests are to be scheduled immediately, leading to a bufferless system model: if at job arrival no free server is found, the job is lost. Backed by real world Grid measurements [Christo07], we will assume Poisson job arrivals (mean arrival rate λ_i at site *i*). This implies that, given the lack of buffers, we can use the *ErlangB* formula (1) to calculate the total number of server CPUs *n* required to achieve a maximal loss rate *L*.

$$L = \text{ErlangB}(n, \lambda, \mu) = \frac{(\lambda/\mu)^n / n!}{\sum_{k=0}^{n} (\lambda/\mu)^k / k!} \quad (1)$$

To place the *n* server CPUs among the *K* (≤ *N*) server site locations chosen in step 1, we consider three strategies:

(i) *unif*: uniformly distribute the server CPUs among all K server sites: $n_k = n/K$, for each server location k = 1..K.

(ii) *prop*: distribute the server CPUs proportionally to the (cluster) arrival rate at each server site k: $n_k = \lambda_k^*/(K \cdot \lambda)$, with $\lambda = \Sigma\, \lambda_i$ and $\lambda_k^* = \Sigma\, \lambda_i \cdot S_{ik}$, where S_{ik} is 1 if and only if k is the server site closest to i (i.e.

as defined in the ILP of Fig. 2). Note that λ_k* equals the total job arrival rate summed over all Grid sites in cluster k.

(iii) *lloss*: try and achieve the same "local loss rate" at each server site, i.e. use ErlangB to calculate nk* as the number of server CPUs to install locally at server site k to achieve loss rate L (i.e. solve L = ErlangB(n_k*, λ_k*, μ)) and install n_k = n· n_k*/(Σ n_i*) servers.

Intuitively, we expect the prop and lloss strategies to perform better than unif, since more server capacity will be installed where more traffic is arriving. Case studies below will assess the penalty of using the simple unif strategy in terms of required network resources.

C. Determining the inter-site bandwidths

Once the location and the amount of CPUs are fixed, only the Grid scheduling algorithms determine the amount of jobs, and hence bandwidth, that will be exchanged between each Grid (site, server)-pair. To demonstrate that the difference may indeed be substantial, we will consider three scheduling alternatives. Note that all algorithms account for the anycast routing principle: it depends on the instantaneous availability of Grid resources, without a priori decision on where to execute a job. Since we are interested in minimizing the required amount of resources, including link bandwidth, all scheduling strategies considered will always try to use a 'local' server CPU before anything else. Jobs arriving at a site i belonging to cluster k (as derived in step 1), with m_k as cluster center (i.e. server site) will always be scheduled on a 'local' free server CPU at site m_k if available. The strategies only differ in choosing an alternative CPU if for a job arriving at i, all CPUs at its cluster center m_k are occupied:

(i) *rand*: randomly choose a free server CPU (i.e. among F free servers, each has 1/F chance);

(ii) *SP*: the closest free server in terms of routing distance (H_{ij} as defined in the algorithms in Section 2.2.1.1.A, e.g. hop count) is chosen, thus striving to minimize network usage;

(iii) *mostfree*: choose a free CPU at server site f, where f is the server site with the highest number of free server CPUs, in an attempt to avoid overloading sites and thus limiting non-local job execution.

To calculate the job exchange rate for every Grid (site, server)-pair, we resort to simulations. In this, we make abstraction of the network capacity (since that is exactly what we want to calculate in the next step): we assume infinite link bandwidth, the only way jobs can be lost is because of lack of CPU capacity.

The reason for resorting to simulations is that because of the anycast principle it is hard to obtain accurate estimates for the inter-site traffic bandwidth using analytical techniques. To illustrate this, we compared our simulation results with those obtained using a fixed-point approximation methodology. We iteratively solve the equations (2)- (4), initializing the system with λ_k'=λ_k. Equation (2) gives the blocking probability at site k, i.e. the probability that all its nk servers are occupied. The total job arrival rate λ_k' at site k is calculated from equation (3) as the sum of the locally arriving jobs λ_k and the fractions f_{jk} of the jobs arriving at all other sites j which are blocking there with probability L_j. The fractions f_{jk} of jobs blocking at j and sent to k, are assumed to be proportional to the probability 1–L_k of finding a free server at site k, as given in equation (4). We stop the numerical iterations solving this system of equations when the difference between successive calculations of the loss rates with equation (2) is smaller than a given tolerance τ (in the results below, we set τ = 10^{-8}).

$$L_k = \text{ErlangB}\left(n_k, \lambda'_k, \mu\right) \quad (2)$$

$$\lambda'_k = \lambda_k + \sum_{\substack{j=1 \\ j \neq k}}^{K} L_j \cdot \lambda_j \cdot f_{jk} \quad (3)$$

$$f_{jk} = \frac{1 - L_k}{\sum_{\substack{m=1 \\ m \neq j}}^{K} \left(1 - L_m\right)} \quad (4)$$

Note that the approximation lies in fixing a priori the amount of jobs that is off-loaded to a remote site: a blocked job originally intended for site i is sent to a remote site k with probability f_{jk}, regardless of the availability of servers at site k. Hence, under this assumption, the blocking rate for traffic initially sent to site i is given by P_i as in equation (5). The total blocking probability P, as given by equation (6), will be larger than the ErlangB blocking of equation (1) achieved by fully sharing all available server capacity, and sending jobs using the anycast principle to any free server.

$$P_i = L_i \cdot \left(1 - \sum_{\substack{k=1 \\ k \neq i}}^{K} f_{ik} \cdot \left(1 - L_k\right)\right) \quad (5)$$

$$P = \frac{\sum_i \lambda_i \cdot P_i}{\sum_i \lambda_{ij}} \quad (6)$$

D. Determining the link bandwidths

After the previous step, we know how many jobs/s are exchanged between every Grid node pair in the considered Grid network. Assuming a given data size distribution, the number of jobs/s can be translated into a bandwidth requirement (e.g. in Mbit/s). Thus, we now have obtained a traffic matrix listing the traffic demand between each (source, destination)-pair. Hence, we can apply ‘classical’ network dimensioning algorithms to assess the required network resources (e.g. number of wavelengths on each link) for given switching technologies (e.g. OBS, OCS or hybrid [VanBreuse06]) and requirements (e.g. resilience).

For the case study presented next, we will assume shortest path routing and make abstraction of the actual network technology to judge the cited scheduling and CPU dimensioning variants by. We will use the average hop count traversed by a job as a measure of network resource usage. Note that by accounting for overhead (cf. packet headers or burst header packet offsets), or discrete bandwidth increments (multiples of wavelength bandwidth), the actual difference in bandwidth requirements can be larger than what the hop count based results may suggest. Yet, the qualitative conclusions will remain valid.

2.2.1.2 *Case study*

In the previous section, we have outlined a step-wise scheme to dimension both server CPU and network resources for computational Grid. We will now apply it in a realistic case study, to highlight the importance of choosing an appropriate scheduling and server CPU placement algorithm when trying to limit the network resource requirements. The scenario and input parameters are outlined and motivated in Section 2.2.1.2A, whereas the actual results will be subsequently discussed.

A. Scenario

To obtain a realistic case study, we performed measurements on a real world Grid, deployed in Europe in the frame of the Large Hadron Collider (LHC) experiments in Geneva and the Enabling Grids for E-sciencE (EGEE) project [EGEE], referred to in short as the EGEE/LCG Grid. From Grid-wide job arrival logs, it was derived that the Poisson traffic model (with negatively exponentially distributed inter-arrival times) accurately fits the real world arrivals [Christo07].

We considered two network topology and job demand rate cases, whose topologies are sketched in Figure 3, and the input parameters are summarized in Table 1. The first used a fairly densely meshed European backbone network (the “Large Topology” taken from [Maesschalck03]), with artificially generated job arrival rates at each site (each rate λ_i was with 30% chance uniformly chosen in [1,15] and 70% from [30,60]).

The second case is based on measurement data from the EGEE/LCG Grid (the same data set as [Christo07]), and for a topology based on the EGEE site locations and the Geant2 [Geant2] network topology and its associated various national research and education networks (NRENs). The arrival rates at each site were set to the values derived from a one-month trace file. (Note that the job trace data comprised 58 Grid sites, rather than the 20 of the Geant2-inspired topology: we attributed their job arrivals to the geographically closest Geant2 site, based on the coordinates of the EGEE/LCG sites as found with GeoIP [GeoIP].) In this EGEE/LCG case, also the average job duration was derived from the real-life trace file.

For both cases, we applied the dimensioning strategy outlined in Sections 2.2.1.1.A-C. As acceptable job loss for the ErlangB calculation we chose L = 0.05.

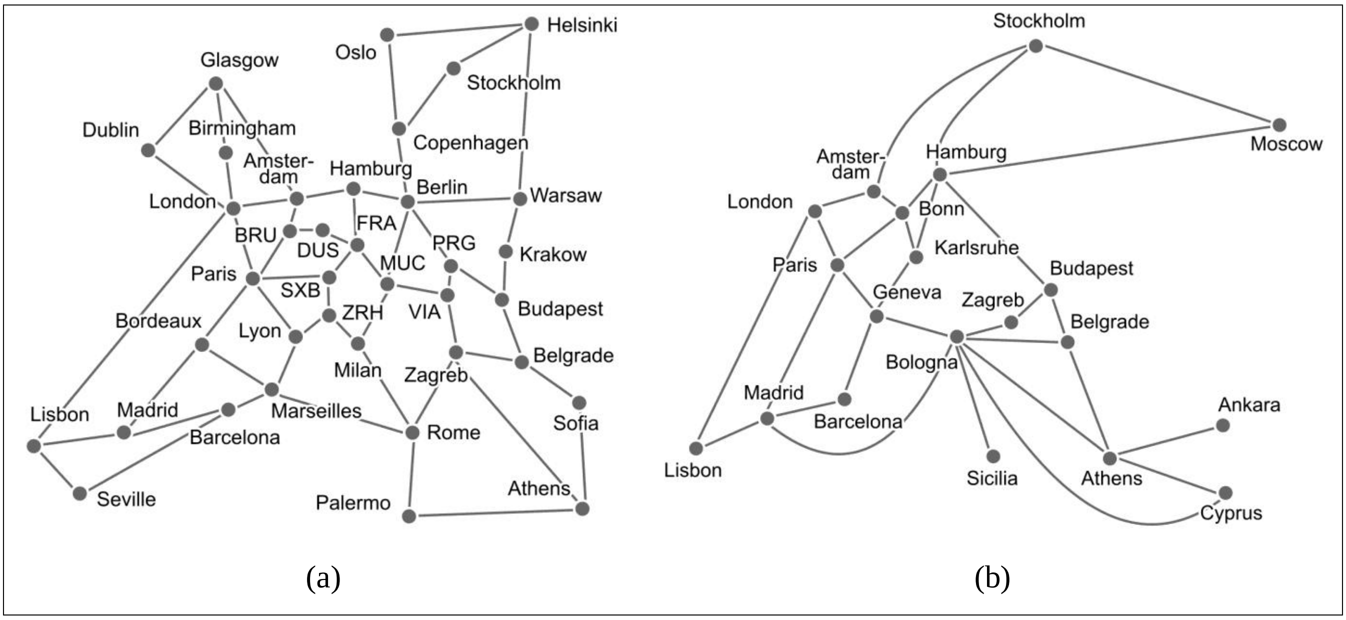


Figure 3: Case study topologies: (a) a European backbone network, (b) Geant2 network.

Parameter	*Case 1 (EU)*	*Case 2 (EGEE/Geant2)*
Topology	European backbone [Maesschalck03]	Geant2 network
Number of nodes	37	20
Number of links	57	31
Average shortest path hop count	3.62	2.55
Job duration	100 s (per job)	854.92 s (per job)
IAT distribution	Exponential distribution	Exponential distribution
Average arrival rate over all sites	2.23E–01 jobs/s	3.56E–02 jobs/s
Stdev of arrival rate over all sites	2.02E–01 jobs/s	6.47E–02 jobs/s
Total arrival rate over all sites	8.24E+00 jobs/s	7.12E–01 jobs/s

Table 1: Case study parameters

B. Local processing rates

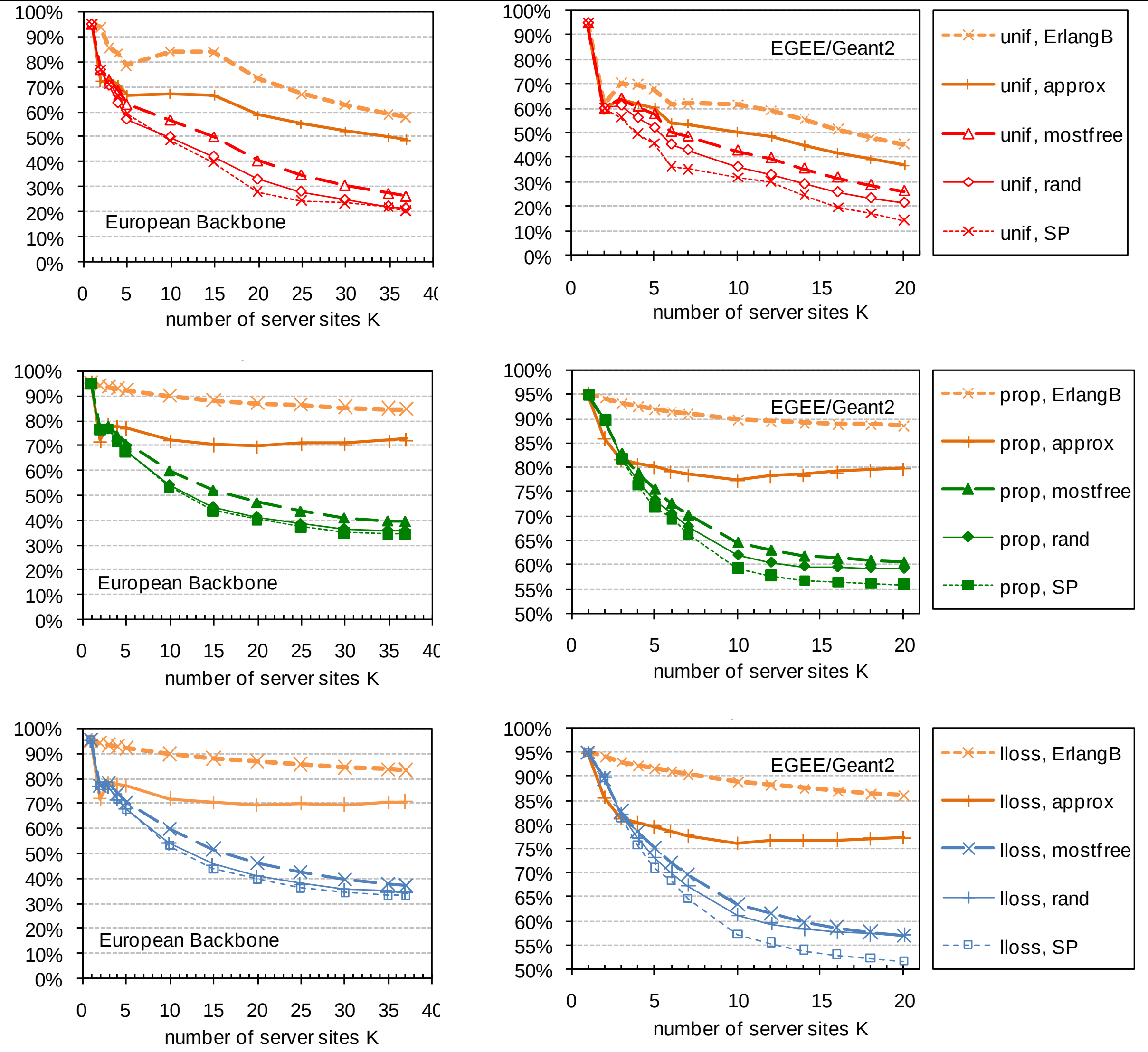
To evaluate the CPU distribution and job scheduling algorithms, a first criterion we considered was how many jobs are off-loaded to a remote site, or its complement, the so-called ‘local processing rate’: the fraction of jobs that is processed at the closest server. To limit network load, we strive for keeping the local processing rate high. This fraction of jobs that is executed at their respective closest server site is plotted in Fig. 4 against the number of chosen server site locations K. Note that the maximal value of 95% is due to the L = 0.05 target job loss rate we dimensioned the server sites for.

To assess the influence of offloading jobs to other sites, we have also calculated an upper bound for the local processing rate, using the ErlangB formula. This bound was obtained by assuming that jobs only may be executed at the closest server site (see the simplifying assumption used in the server site selection approach of Section 2.2.1.1.A). Given this assumption, the maximal local processing rate at server site k equals 1-L_k, with L_k = L calculated using ErlangB formula (1) for N = n_k the number of servers at site k, and $\lambda = \lambda_k$ the aggregate arrival rate of its closest Grid sites.

With respect to the server distribution schemes, placing more servers where more jobs originate is beneficial: the prop and *lloss* strategies attain higher local processing rates than *unif*. The difference between prop and *lloss* from this respect is minimal (relative differences less than 5%): only for the European backbone case we noted a slightly higher local rate for *lloss* for low server site counts K, in all other cases prop attains a higher fraction of jobs executed at the respective closest server site. This indicates that the more complex *lloss* dimensioning strategy doesn’t seem to pay off compared to the simple prop strategy.

From the graphs it is clear that the scheduling algorithm also impacts the local processing rate. Among the considered alternatives, *mostfree* from this perspective performs best.

With respect to the influence of the traffic and network topology, we note that in the EGEE/Geant2 case the matching the server capacities to the traffic arriving (prop and lloss cases) seems more effective: especially for larger server counts K, the local processing rate stabilizes around 55-65% in the EGEE/Geant2 case (compared to 30-40% for the EU case). This can be explained by the larger variance in job arrival rates in the EGEE/Geant2 case (see Table 1): for bigger discrepancies in site arrival rates, the server counts will differ more between the unif and prop/lloss dimension strategies.



(a) (b)

Figure 4: The fraction of jobs executed at the respective closest server site ('local processing rate') is maximized by intelligently positioning server capacity (prop vs unif), and is well below the ErlangB upper bound: results for (a) the European backbone network, (b) the Geant2 network. Note the different Y-axis scale in the prop and lloss graphs for EEGE/Geant2. The analytical fixed point approximation (approx) fails to match the more accurate simulation results for a higher amount of server sites (K > 5).

Looking at the ErlangB upper bound, the better performance of the prop/lloss approaches (compared to unif) also is immediately apparent. It is striking that there is a rather big gap between the ErlangB upper bound, and the actually attained local processing rates. This suggests that as soon as the total amount of servers is distributed over multiple locations, there is a non-negligible amount of jobs that is sent to remote sites (instead of being dropped as in case of the ErlangB bound assumptions), which there compete for server capacity with the locally arriving jobs. Yet, by allowing this interchange of jobs, the overall success rate is improved: the target success rate attained by non-local job execution remains at 95%, whereas the ErlangB curve drops well below that (note that in case of ErlangB the 'local processing rate' shown in Figure 4 is exactly the success rate since it assumes no remote execution of jobs).

Note also that the 'approx' curves, showing the results of the fixed point approximation solving the equations (2)-(4), do not very well match the simulation results. This approach indeed does not accurately model the site blocking probabilities' inter-dependencies. As a result, the approximation overestimates local processing probabilities as soon as the number of Grid server sites K increases (see curves for K > 5).

Solely looking at local processing rates, one may be tempted to opt for a minimal number of server locations. In the following sections we will argue this is not optimal from a correct network perspective.

C. Used link bandwidth

From network perspective, the most important criterion is network bandwidth. To establish the network dimensions in the considered case study, we would need to choose a particular network technology (OBS versus OCS, or hybrids). Yet, these relate to the traffic matrix stating the amount of bandwidth exchanged between each node pair. A useful measure to comprehensively summarize this information in the assumed Grid context is the average hop count a job needs to traverse to reach the server it will be executed on. Hence, we will use the average job hop count as a measure to judge the network capacity requirements. We obtained this measure from the simulation approach described in Section 2.2.1.1.C. We summarize the average job hop count results for varying number of server sites K in Figure 5.

These graphs also include results obtained from the analytical fixed point approximation. Given the solution of the equations (2)-(4), the bandwidth B_{ik} flowing from site i to server site k can be calculated by equation (7). From these values B_{ik} the average hop count can be calculated as from the simulation results.

$$B_{ik} = \begin{cases} \lambda_i \cdot (1 - L_k) & k = \text{closest server c for site i} \\ \lambda_i \cdot L_c \cdot f_{ck} \cdot (1 - L_k) & k \neq \text{closest server for site i (c = closest server for site i)} \end{cases} \qquad (7)$$

Comparing the analytical approximation, we note a mismatch compared to simulation results for larger values of number of server sites K, as before. For the unif site dimensioning strategy results in case of EGEE/Geant2, we observe non-smooth fluctuations. The reason is that in the unif case, all server sites get an equal portion of the available server capacity, while there is quite some discrepancy in job arrival rates. Hence, adding extra server sites may result in a quite drastic change in the inter-site traffic rates (see also the less smooth 'local processing rate curves' in Section 2.2.1.2.B), stemming from a severe reduction in server capacity in certain network segments. The fact that this is far less pronounced in the European Backbone case can be explained by the smaller variation in job arrival rates. In the prop and lloss cases, the server capacities better match the arrival rates and hence the curves evolve smoothly.

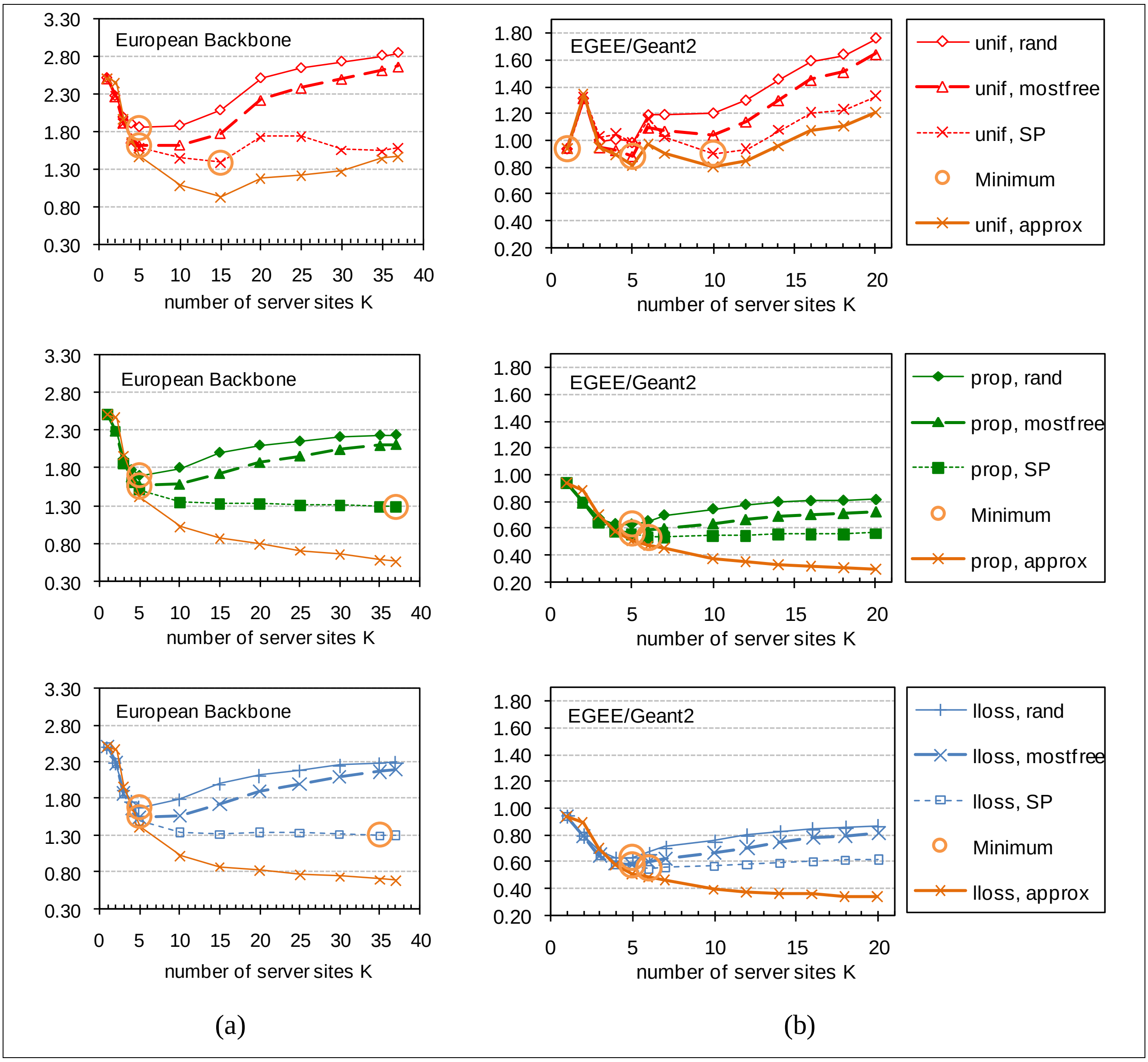


Figure 5: The required network capacity, which is proportional to the average job hop count, is minimized by adopting shortest path routing and scheduling (SP), intelligently positioning server capacity (prop vs unif), and deploying a reasonable number of server locations: average hop count over all jobs for (a) the European backbone network, (b) the Geant2 network. The analytical fixed point approximation (approx) deviates from simulation results because it does not accurately capture site inter-dependencies.

Comparing the various combinations of dimensioning and scheduling alternatives, the relative influence of the scheduling algorithm seems to be important. The reasonably large fraction of traffic sent to non-closest server sites—recall the 'local processing rates' from the previous section—has an important influence on the network load. Hence, by adopting shortest path driven scheduling (SP), the lowest average job hob count is reached.

The influence of the dimensioning strategy is also obvious, though less significant. Especially for a larger number of server sites K, it pays off to intelligently distribute server capacity: prop and lloss (which hardly differ in resulting average job hop count) result in lower network load than straightforward uniform server distribution.

Comparing the European backbone case with the EGEE/Geant2 case, we note that the major qualitative difference lies in the curves for the uniform dimensioning strategy: this curve is mainly increasing for larger values of K in the EGEE/Geant2 case. This can be explained by the larger relative variations in shortest path hop counts in this network: the penalty of unintelligently distributing server capacity is more pronounced.

With respect to the choice of the number of server sites K, we observe there is an optimal choice, which tends to lie around K = 5 in the studied cases, ranging between 1/7 and 1/4 of the total number of sites. Note that the optimum depends on both the dimensioning strategy and the scheduling approach. When too much server sites are installed, the total server capacity is fragmented too much, resulting not only in low 'local processing rates' (see Figure 4) but also a lot of jobs sent to remote servers. Indeed, for larger number of server sites, the opportunities of statistical multiplexing diminish and with it the probability of finding a free server at the closest server site for a particular job. This apparently outweighs the fact of lowering the average distance to a server site.

Contrary to (rather scarce) earlier work on Grid dimensioning, we proposed a dimensioning methodology fully taking into account the anycast routing principle, i.e. without presuming a priori knowledge of (source, destination)-based traffic. The proposed step-wise methodology is suitable for dimensioning both server and network capacities. We outlined it for computational Grids, but extension to also incorporate e.g. storage capacity is possible.

We used the methodology to evaluate various scheduling algorithms and server dimensioning options with respect to the required network capacity. From two case studies on European topologies, we concluded that placing server capacity where a lot of jobs arrive is important to minimize network bandwidth requirements: the prop dimensioning strategy, placing a number of servers proportional to the job arrival rates at its closest Grid sites is most beneficial. With respect to Grid scheduling, a simple shortest path (SP) strategy, preferring closer server sites, led to the lowest bandwidth demands. With respect to choosing an appropriate number of server sites K (ranging from 1 to the total number of Grid sites N), we found that there is an optimal value. For a larger number of server sites, the total server capacity gets fragmented, reducing opportunities for statistical multiplexing, whereas for smaller server site counts K the average distance jobs need to travel is too large. That optimum of K depends on the scheduling algorithm and server site dimensioning strategy, and in the considered case studies was about 1/7 to 1/4 of the total number of sites.

2.2.2 Resource Dimensioning

In this section, we propose a task scheduling algorithm that minimizes the number of computational resources required for task scheduling (Issue 1 – Section 1.1), while increasing the utilization efficiency and the percentage of tasks served by the Grid without violating their QoS requirements (Issue 4 – Section 1.1) [Doulamis08]. The tasks QoS requirements are given in the form of a desired start and finish time. More specifically the proposed scheduling scheme exploits concepts derived from spectral clustering, and groups tasks together for assignment to a computing resource so as to (a) minimize the time overlapping of the tasks assigned to a given resource and (b) maximize the time overlapping among tasks assigned to different resources. The above two objectives are transformed into a matrix representation problem. The proposed scheduling scheme uses the notions of generalized eigenvalues and the Ky-Fan theorem to obtain an algorithm of polynomial order. The main performance metric of the proposed algorithm is the minimum number of computational resources it requires in order to schedule all the tasks without violating their QoS requirements. Experimental results show that the proposed algorithm outperforms a greedy algorithm as well as other previously proposed scheduling schemes for different values of the granularity and the load of the submitted tasks.

We also examine a task scheduling and data migration problem for Grid Networks, which we refer to as the Data Consolidation (DC) problem [Kokkinos08]. DC arises when a task needs for its execution multiple pieces of data, possibly scattered throughout the Grid Network. In such a case, the scheduler and the data manager must select (i) the data replicas to be used, (ii) the site where these data will accumulate for the task to be executed, and (iii) the routing paths to be followed. The algorithms or policies for selecting the data replicas, the data consolidating site and the corresponding paths comprise a Data Consolidation scheme. The investigation of the DC problem relates strongly to the placement of the storage resources and the datasets in the Grid Network (Issues 2 – Section 1.1). We propose and experimentally evaluate a number of DC schemes, and examine their performance by altering various parameters such as the number of data sets, the number of storage resources, the data requirements of the task, etc (Issue 4 – Section 1.1). Our simulation results brace our belief that DC is an important problem that needs to be addressed in the design of Data Grids. If Data Consolidation is performed efficiently, benefits are provided in terms of task delay, network load and other performance related parameters.

The Phosphorus' emphasis is on network-related aspects of Grid computing, and thus its focus is on data-intensive applications that are heavily dependent on communication resources. The Phosphorus testbed topology is an established infrastructure where issues like quantity and placement of resources has already been decided and implemented. So, we examine algorithms that maximize resource usage and depend heavily on data placement.

More specifically, the DC schemes seem to fit quite well with the various Phosphorus applications (e.g., KoDaVis, INCA, WISDOM) [D.3.1], where fragments of data are sent to a central location for processing. Furthermore, the proposed task scheduling algorithm that minimizes the number of processors required for task scheduling, while increasing the utilization efficiency and the percentage of tasks served by the Grid, could be used in the WISDOM application. Finally, we can argue that the same concepts and ideas used by the proposed algorithm in order to find the minimum number of computational resource required for task execution could also be used for finding the minimum number of storage resources needed for serving the large storage needs of the Phosphorus related applications.

2.2.2.1 *Spectral Clustering Scheduling (SCS) Algorithm*

We present an algorithm that finds the minimum number of computational resources required and the corresponding task-to-resource assignments, so as to serve the tasks with no violations of their constraints. In general, a number of task-to-resource scheduling algorithms have been proposed that try to maximize overall system performance (resource utilization efficiency), while missing to satisfy users requirements and vice versa. The algorithm we propose takes into account both considerations assigning tasks to resources so that a) the time overlapping between tasks assigned to the same resource are minimized, while simultaneously b) overall Grid utilization efficiency is maximized. The proposed algorithm will be referred to as the Spectral Clustering Scheduling (SCS) scheme, since its basic algorithmic components stem from spectral clustering.

The SCS algorithm can help the capacity planning of the Grid infrastructure since it allows advance estimation of the number of computational resources required in order to satisfy the requested tasks' requirements. This becomes more relevant with the emergence of cloud computing. Knowing in advance the number of resources needed to satisfy the requirements of the users, could be useful in capacity planning and in achieving predictability in such systems.

We perform a number of experiments, in which we compare SCS with other known algorithms. In the case where no violations of the task requirements are permitted, we increase the number of computational resources until the constraints of all the tasks are satisfied, and we compare the minimum number of computational resources required to achieve that, for the different algorithms examined.

Task Requirements and Overlapping

We define task requirements through the task start and finish times. In this case, the Grid is aware of the actual time period during which the task will occupy a computational resource. There is a direct relation between the tasks' time constraints and the time the task should reserve for its execution, which often also relates to the price the user will be charged. For example cloud computing services, such as the Amazon EC2, price the virtual computing environments the users create, per hour of use. Also, precedence constraints among tasks can be accounted for using task start and finish times. Such an approach is similar to the way humans use to hire consumable resources in their daily routines. For example, when we book a room in a hotel we define our arrival (start) and departure (finish) date (time).

Let us denote by $V=\{T_i\}_{i=1,\dots,N}$ the set of tasks that request service in a Grid infrastructure consisting of *M* computational resources. Let us also denote by ST_i the desired *Start Time* of Task T_i and by FT_i its desired *Finish Time*. We assume that the tasks are atomic operations and are scheduled in a non-preemptable, non-interruptible way. A task is said to be non-preemptable if once it starts execution on a resource, it has to be completed on that resource. Additionally, a task is said to be non-interruptible if once it starts execution it cannot be interrupted by other tasks and resume execution later. Under this assumption, if a task has been assigned for execution on a resource and another task requests service on an overlapping time interval, then, the second task should either be reassigned to another available resource, or have to undergo a violation of its start or finish time, or both of them. Minimizing the number of computational resources used, or equivalently maximizing their utilization efficiency is a primary objective of the algorithm we will propose. Eliminating or minimizing violations of the task start and finish times is our second important objective.

In a scheduling policy with fixed (as opposed to malleable) reservation intervals, it is probable for two tasks to request service in the same time interval, resulting in a task overlapping or a time conflict. If the overlapping tasks are assigned to the same Grid resource, one of them will suffer a violation of its requirements and will not be executed within its specified time interval. The demand for minimizing such violations can be expressed as the minimization of the time overlapping among the tasks assigned to the same resource. From the users' point of view, the Grid should have enough resources (ideally infinite) to guarantee that all tasks are feasibly executed without violations of their requirements. However, from the Grid service providers' point view, an infrastructure with too many resources is wasteful when the resources are active only occasionally. Ideally, all computational resources should be busy at all times to yield the maximum benefit of the infrastructure. This demand can be expressed as the maximization of the overlapping degree among the time intervals of the tasks assigned to different resources.

A metric that will be useful in the Grid scheduling algorithm we will propose is the degree of overlapping (or equivalently of non-overlapping) between two tasks. In particular, we denote by σ_{ij} the non-overlapping measure between tasks T_i and T_j:

$$\sigma_{ij} = \begin{cases} a, & \text{if } T_i, T_j \text{ are non-overlapping in time} \\ 0, & \text{if } T_i, T_j \text{ overlap in time,} \end{cases} \tag{1}$$

where $\alpha > 0$ is a positive non-zero constant.

The non-overlapping measure σ_{ij} takes zero values when tasks T_i and T_j overlap in time (at least during some time interval) and positive non-zero values when they do not. We have adopted this binary non-overlapping measure, even though other definitions of the non-overlapping measure could also be used to permit a more flexible description of task requirements violations.

The duration of task T_i is expressed as the difference $d_i = FT_i - ST_i$. We also define the average duration *D* of the tasks that have to be scheduled as

$$D = \frac{\sum_{i=1}^{N} d_i}{N}. \tag{2}$$

N is the number of tasks requesting service in the Grid within a time interval *T*, where *T* represents the time between two successive executions of the scheduling algorithm. The scheduler considers the tasks that arrive within a time horizon *T* and selects the most appropriate resources for executing them, trying to minimize violations of their timing constraints. The task durations d_i, *i*=1,2,..,*N*, may be similar for all tasks or vary significantly from task to task. The first will be referred to as the symmetric task case, while the second as the asymmetric case.

Joint Optimization of Resource Performance and Requirements

The scheduler has to find a mapping of the pending tasks to the available resources. Let us denote by C_r the set of tasks assigned for execution on computational resource *r*. The sets C_r, *r*=1,2,…,*M*, for different resources *r* are mutually exclusive, since a task cannot be split and executed on different resources (non-interruptible assumption).

As stated previously, an efficient scheduling scheme should assign all the *N* pending tasks to the *M* available computational resources so as to a) maximize the overall utilization of the resources used, while simultaneously

b) minimize the tasks requirements violations. The first requirement is met when the task overlapping among different resources is maximized; in that case, the utilization of all resources in the Grid will be as high as possible, so that resources do not stay idle much of the time. The second requirement implies that the tasks assigned to a given resource should exhibit minimal overlapping.

In our formulation, the two above mentioned requirements are expressed as a) the minimization of the non-overlapping measure among tasks assigned to different resources and b) the maximization of a non-overlapping measure among all the tasks assigned to the same computational resource. Using the non-overlapping measure σ_{ij} between tasks, we express the non-overlapping degree of the tasks assigned to computational resource *r* and those that are not as the sum of the pair wise non-overlapping measures between the tasks in *r* and those that are not assigned to *r*, normalized over the sum of non-overlapping measures between tasks in *r* and all tasks in the Grid, that is,

$$P_r = \frac{\sum_{i \in C_r, j \notin C_r} \sigma_{ij}}{\sum_{i \in C_r, j \in V} \sigma_{ij}}. \tag{3}$$

Low values of P_r in Eq. (3) indicate that many other resources in the Grid are simultaneously active with resource *r*. When P_r is large, the number of resources that are simultaneously active with *r* is small, indicating that most of the remaining resources are idle.

Similarly, the the non-overlapping degree of all tasks assigned to resource *r*, is expressed as,

$$Q_r = \frac{\sum_{i \in C_r, j \in C_r} \sigma_{ij}}{\sum_{i \in C_r, j \in V} \sigma_{ij}}. \tag{4}$$

The denominator of Eq. (4) expresses the non-overlapping degrees of the tasks mapped to resource *r* with all the *N* available tasks, including the ones assigned to *r*, and it is used for normalization purposes. Otherwise, optimizing only the numerator of (4) would favour the trivial solution of one task per resource. Parameter Q_r expresses a measure of the overall requirements violation for the tasks assigned to the r^{th} resource. As Q_r increases, task violations decrease for resource *r*, since fewer tasks overlap in time on that resource.

Considering all the *M* resources of the Grid, we can define a measure a measure *P* for the overall resource utilization as

$$P = \sum_{r=1}^{M} P_r \tag{5}$$

and a measure *Q* for the total tasks' QoS violation as

$$Q = \sum_{r=1}^{M} Q_r \tag{6}$$

As stated previously, a scheduler that tries to make efficient use of the resources, while also meeting user requirements, should minimize P and simultaneously maximize Q. This would result in maximal Grid resource utilization and minimal task requirements violations. However, it is clear that

$$P + Q = M. \tag{7}$$

Equation (7) shows that the minimization of P (equation (5)) simultaneously yields maximization of *Q* (equation (6)) and vice versa. Hence, in our problem, the two aforementioned optimization objectives require in fact the use of identical means and they can be met simultaneously. This is intuitively satisfying, since scheduling a set of tasks in a way that makes efficient use of resources is also expected to help meet the requirements of the set of tasks that are scheduled. Therefore, it is enough to optimize only one of the two criteria. In our case, and without loss of generality, we select to minimize *P*, by trying to find the task assignment on the *M* computational resources (that is, a partitioning C_r) that minimizes

$$\hat{C}_r : \min P = \min \sum_{r=1}^{M} \frac{\sum\limits_{i\in C_r, j\notin C_r} \sigma_{ij}}{\sum\limits_{i\in C_r, j\in V} \sigma_{ij}} \text{, for all } r=1,\ldots,M, \tag{8}$$

where $\hat{C}_r$ is the set of tasks assigned to resource *r* that minimizes Eq. (8).

The Scheduling Algorithm

Optimizing Eq. (8) is actually a NP-complete problem. Even for the case of *M*=2 resources, it remains NP-complete, and is practically intractable when the number of tasks is large. However, we can overcome this difficulty by transforming the problem of Eq. (8) into a matrix based representation. Then, an approximate solution in the discrete space can be found using concepts derived from eigenvalue analysis.

Matrix Representation

Let us denote by $\mathbf{\Sigma} = [\sigma_{ij}]$ the matrix containing the non-overlapping measures σ_{ij} for all *N*x*N* pairs of tasks T_i and T_j. Let us now denote by $\mathbf{e}_r = [\cdots e_r^u \cdots]^T$ a *N*x1 indicator vector whose uth entry is given by

$$e_r^u = \begin{cases} 1, & \text{if task } T_u \text{ is assigned to processor } r \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Vector $\mathbf{e}_r$ indicates which of the *N* tasks are executed on resource *r*; indices of tasks assigned to resource *r* are marked with one, and the remaining indices with zero. Since, the Grid infrastructure consists of *M* resources, *M* different vectors $\mathbf{e}_r$, *r*=1,2,…,*M* are defined, each indicating the tasks assigned for execution on a given resource. Therefore, the optimization problem of (8) is equivalent to finding the optimal indicator vectors $\hat{\mathbf{e}}_r$, for all *r*, that minimize Eq. (8). Consequently, Eq. (8) can be written as

$$\hat{\mathbf{e}}_r, \forall r : \min P = \min \sum_{r=1}^{M} \frac{\sum\limits_{i\in C_r, j\notin C_r} \sigma_{ij}}{\sum\limits_{i\in C_r, j\in V} \sigma_{ij}} . \tag{10}$$

A difficulty in optimizing (10) is that its right hand side is not expressed as a function of the indicator vectors $\mathbf{e}_r$. Thus, we need to re-write the right part of Eq. (10) in a form that includes the vectors $\mathbf{e}_r$. To do so, we denote by $\mathbf{L} = diag(\cdots l_i \cdots)$ the diagonal matrix, whose elements l_i, i=1,2,...,*N*, express the cumulative non-overlapping degree of task T_i with the remaining tasks, that is,

$$l_i = \sum_j \sigma_{ij}. \tag{11}$$

Using matrices *L* and *Σ* ($\mathbf{\Sigma} = [\sigma_{ij}]$), we (11) as a function of the vectors $\mathbf{e}_r$. In particular, we have

$$\mathbf{e}_r^T(\mathbf{L} - \mathbf{\Sigma})\mathbf{e}_r = \sum_{i \in C_r, j \notin C_r} \sigma_{ij} . \tag{12}$$

In a similar way, the denominator in (10) is related to the indicator vector $\mathbf{e}_r$ as follows

$$\mathbf{e}_r^T \mathbf{L} \mathbf{e}_r = \sum_{i \in C_r, j \in V} \sigma_{ij}. \tag{13}$$

Using (12) and (13), we can re-write (10) as

$$\hat{\mathbf{e}}_r, \forall r : \min P = \min \sum_{r=1}^{M} \frac{\mathbf{e}_r^T(\mathbf{L} - \mathbf{\Sigma})\mathbf{e}_r}{\mathbf{e}_r^T \mathbf{L} \mathbf{e}_r} . \tag{14}$$

Equation (14) yields the optimal vectors $\hat{\mathbf{e}}_r$, that is, the task-to-resource assignments that maximize resource utilization and, consequently, also minimize the non-overlapping degree among all resources [see Eq. (7)].

Optimization in the Continuous Domain

As already mentioned, we assume non-interruptible tasks that have to be completed on the resource they start execution on. This implies that the indicator vectors $\mathbf{e}_r$ should take binary values; one, for tasks executed on resource *r*, and zero for other tasks. In other words, if we form the indicator matrix $\mathbf{E} = [e_1 \cdots e_M]$, the columns of which correspond to the *M* computational resources in the Grid and the rows to the *N* tasks, then the rows of *E* have only one unit entry and the remaining entries are zero. Matrix *E* can be seen as a binary mask that indicates the resource each task is mapped to.

Optimization of (14) under the discrete representation of the indicator matrix *E* (or equivalently the indicator vectors $\mathbf{e}_r$) is still a NP hard problem. However, if we relax the indicator matrix *E* to take values in the continuous domain, we can solve the problem in polynomial time. We denote by $\mathbf{E}_R$ the relaxed version of the indicator matrix *E*, i.e., a matrix whose rows take real instead of binary values. Then, we discretize the continuous values of the relaxed matrix $\mathbf{E}_R$ to get an approximate solution of the scheduling problem.

It can be proven that the right part of Eq. (14) can be rewritten as

$$P = M - trace(\mathbf{Y}^T \mathbf{L}^{-1/2} \mathbf{\Sigma}\, \mathbf{L}^{-1/2} \mathbf{Y})\,, \tag{15a}$$

subject to

$$\mathbf{Y}^T \mathbf{Y} = \mathbf{I}\,, \tag{15b}$$

where *Y* is a matrix that is related to matrix $\mathbf{E}_R$ through the following equation

$$\mathbf{L}^{-1/2} \mathbf{Y} = \mathbf{E}_R \mathbf{\Lambda}\,. \tag{16}$$

In (16), Λ is any arbitrary $M \times M$ matrix. In this work, we select *Λ* to be equal to the identity matrix, *Λ*=*I*. Then, the relaxed indicator matrix $\mathbf{E}_R$, which is actually the matrix we are looking for, is given as

$$\mathbf{E}_R = \mathbf{L}^{-1/2} \mathbf{Y}\,. \tag{17}$$

Minimization of (15a) is obtained through the Ky-Fan theorem [Veselic03]. The Ky-Fan theorem states that the maximum value of $trace(\mathbf{Y}^T \mathbf{L}^{-1/2} \mathbf{\Sigma}\, \mathbf{L}^{-1/2} \mathbf{Y})$ with respect to the matrix *Y*, subject to the constraint $\mathbf{Y}^T \mathbf{Y} = \mathbf{I}$, is given by the sum of the *M* (*M*<*N*) largest eigenvalues of matrix $\mathbf{L}^{-1/2} \mathbf{\Sigma}\, \mathbf{L}^{-1/2}$. Thus,

$$\max_{subject\ to \mathbf{Y}^T \mathbf{Y} = \mathbf{I}} \{trace(\mathbf{Y}^T \mathbf{L}^{-1/2} \mathbf{\Sigma}\, \mathbf{L}^{-1/2} \mathbf{Y})\} = \sum_{i=1}^{M} \lambda_i\,, \tag{18}$$

where λ_i refers to the *i*th largest eigenvalue of matrix $\mathbf{L}^{-1/2} \mathbf{\Sigma}\, \mathbf{L}^{-1/2}$.

However, the maximization of (18) leads to the minimization of P in (15a), and the minimum value of *P* is

$$\min P = M - \sum_{i=1}^{M} \lambda_i\,. \tag{19}$$

The Ky-Fan theorem also states that this minimum value of *P* is obtained for the matrix

$$\mathbf{Y} = \mathbf{U} \cdot \mathbf{R}, \tag{20}$$

where *U* is a $N \times M$ matrix whose columns are the eigenvectors corresponding to the *M* largest eigenvalues of matrix $\mathbf{L}^{-1/2} \mathbf{\Sigma}\, \mathbf{L}^{-1/2}$ and *R* is an arbitrarily rotation matrix (i.e., orthogonal with determinant of one). Again, a simple approach is to select *R*=*I*, in which case,

$$\mathbf{Y} = \mathbf{U}\,. \tag{21}$$

Therefore, Eq. (15) is minimized at $\mathbf{Y} = \mathbf{U}$ and the minimum value is given by Eq. (17). Then, the optimal values $\hat{\mathbf{E}}_R$ for the relaxed matrix $\mathbf{E}_R$ in the continuous domain will be given as

$$\hat{\mathbf{E}}_R = \mathbf{L}^{-1/2} \mathbf{U}. \tag{22}$$

Equation (22) means that the optimal relaxed matrix $\mathbf{E}_R$ is related to i) the cumulative non-overlapping degree of all tasks and ii) the eigenvectors corresponding to the *M* largest eigenvalues of matrix $\mathbf{L}^{-1/2} \mathbf{\Sigma}\, \mathbf{L}^{-1/2}$.

Discrete Approximation

The optimal matrix $\hat{\mathbf{E}}_R$, given by Eq. (22), does not have the form of the indicator matrix *E* since the entries of $\hat{\mathbf{E}}_R$ are non-integer in general, while *E*'s entries are binary. We recall that a unit entry corresponds to the resource a task is assigned to, under the non-interruptible, non-preemptable assumption. Consequently, the problem is how to round the continuous values of $\hat{\mathbf{E}}_R$ in a discrete form that approximates matrix *E*.

One simple rounding process is to set the maximum value of each row of $\hat{\mathbf{E}}_R$ to be equal to 1 and let the remaining values be equal to 0. However, such an approach yields unsatisfactory performance when there is no dominant maximum value for each row of $\hat{\mathbf{E}}_R$. Furthermore, it handles the rounding process as *N* (equal to the number of rows of $\hat{\mathbf{E}}_R$, or the number of pending tasks) independent problems, implying that each task is scheduled independently of each other. An alternative approach, which is adopted in this work, is to treat the *N* rows of matrix $\hat{\mathbf{E}}_R$ as *M*-dimensional feature vectors. The algorithm clusters the rows of matrix $\hat{\mathbf{E}}_R$ to *M* groups (the number of available resources). The rows of $\hat{\mathbf{E}}_R$ indicate the degree of "fitness" (the association degree) of a task to all the *M* resources. Therefore, the goal of the algorithm is to find the resource to which a task with a specific feature vector fits best.

In particular, we initially normalize the rows of matrix $\hat{\mathbf{E}}_R$ to take values between 0 and 1. Then, we apply the k-means clustering [Stock01] algorithm to these *N* vectors to form the indicator matrix *E*. The *k*-means algorithm consists of three phases, the initialization, the clustering construction, and the updating phase.

Initialization: In this phase, the algorithm arbitrarily selects a set of row vectors of $\hat{\mathbf{E}}_R$ as centers of the clusters to be constructed. The number of clusters is equal to the number of available resources *M*.

Clustering Construction: In this phase, all the remaining vectors of $\hat{\mathbf{E}}_R$ are assigned to the *M* clusters using a metric distance. In particular, a vector is assigned to a cluster by comparing it to all cluster centers and selecting the one whose center is the closest to the chosen vector.

Updating: Following the classification, new centers are created as the average value of all vectors belonging to a cluster. If the new centers are different from the previous ones, a new iteration takes place and the algorithm returns to the clustering construction phase for further processing. If the centers do not change, which implies that the same task assignments have been concluded, no further processing is required and the algorithm terminates.

The performance of the *k*-means algorithm is sensitive to the initial selection of the cluster centers, although it can be shown to always converge to a solution. In this work, to overcome this drawback and simultaneously to search for new possible solutions (approximations of the optimal solution in the discrete domain), we repeat the experiment by selecting each time different vectors in the initialization phase, which, in the sequel, provide different solutions. Among all selections, the minimum is chosen as the closest approximation.

Table I summarizes the steps of the proposed scheduling scheme.

1. Find the non-overlapping measures σ_{ij} for all tasks, using Eq. (1).	*Initialization*
2. Form the respective matrices **L** and **Σ.** **3.** Compute the eigenvectors of matrix $\mathbf{L}^{-1/2}\mathbf{\Sigma}\,\mathbf{L}^{-1/2}$.	*Joint Optimization*
4. Form a matrix **U** of size $N \times M$ whose columns are the *eigenvectors* corresponding to the *M* largest eigenvalues of matrix $\mathbf{L}^{-1/2}\mathbf{\Sigma}\,\mathbf{L}^{-1/2}$.	*Solution in Continuous Domain*
5. Use Eq. (21) to estimate the continuous matrix $\mathbf{E}_R$ as $\hat{\mathbf{E}}_R = \mathbf{L}^{-1/2}\mathbf{U}$. **6.** Normalize the rows of matrix $\mathbf{E}_R$ so as to lie in the interval [0, 1].	*Solution in Discrete Domain*
Set iteration=1, a. While (Iteration<MaxIterations) do b. Arbitrarily select *M* rows of $\mathbf{E}_R$ as centroids of the *M* groups (resources the tasks are assigned to) c. Cluster the remaining rows with respect to the *M* centroids d. Update the groups centroids as the average of the entries of the rows assigned to a particular group e. If new centroids are different from the old ones go to step c. f. Otherwise, Iteration=Iteration+1; go to Step b	*k-means*

Table 2: Main steps of the Spectral Clustering Scheduling (SCS) algorithm.

Scheduling Efficiency

In the previous section, we described an algorithm for mapping tasks with fixed time constraints (start and finish times) to computing resources so that a) the non-overlapping measure of the tasks assigned to different resources is minimized, while b) the non-overlapping measure of the tasks assigned to the same resource is maximized. To evaluate the performance of the algorithm, it is important to compare it to an optimal algorithm: this approach is difficult as scheduling is a NP-hard problem. For this reason, in what follows, we will evaluate the scheduling efficiency using the lower bound on the minimum number of computational resources required for executing the tasks.

We assume that the scheduling algorithm is executed at time intervals of equal duration *T*, and we define the task intensity

$$\lambda = \frac{N}{T}, \qquad (23)$$

as the number of tasks *N* requesting service over the corresponding time interval *T*. A parameter that plays an important role on scheduling efficiency is the task granularity *g*, defined as the ratio of the average task duration *D* over the time horizon *T*:

$$g = \frac{D}{T}. \tag{24}$$

Task granularity is a measure of how large the generated tasks are, compared to the time window *T*. High values of *g* (say, values larger than 0.2) indicate that a task's execution time occupies a significant portion of the time window, and thus, task overlapping is probable. For instance, *g*>0.5 corresponds to an average task duration that is more than half of the time window *T* and, thus, overlapping is almost certain. In contrast, low values of *g* indicate that task execution times are small compared to the time window *T* and thus, we expect better scheduling to be in principle achievable.

As stated the objective is to use the minimum number of resources (thus, achieving maximum utilization per resource) that will provide no task overlapping on each resource (no degradation in the task's requirements). Given the characteristics of the submitted tasks, as expressed by their granularity *g* and task intensity *λ*, a lower bound on the average number of computational resources required for achieving no task overlapping is

$$B = \frac{ND}{T} = N \cdot g \leq \lceil B \rceil \leq M_{opt}, \tag{25}$$

where M_{opt} refers to the minimum number of resources required to achieve no task overlapping by an optimal (exhaustive search) scheduling algorithm. Of course, M_{opt} cannot always be reached in practice, since the exhaustive search is a non-polynomial algorithm.

The lower bound *B* is achievable only in the unlikely case that the tasks require consecutive disjoint intervals within the time horizon *T*; as a result, *B* may be considerably smaller than the actual optimal number of computational resources M_{opt}. As our experimental results will demonstrate, the proposed scheduling scheme results in a number of computational resources that is only a few times larger than the lower bound, and is considerably smaller than that obtained by other conventional approaches.

One measure for evaluating the efficiency of a scheduling algorithm *A* is the utilization factor

$$\rho(A) = \frac{B}{M(A)}, \tag{26}$$

where *M(A)* is the number of resources required for achieving no task overlapping using algorithm *A*. The utilization *ρ(A)* can be viewed as a measure of how far is the number of computational resources required by algorithm *A* to meet all task requirements, from the lower bound *B*.

Since the lower bound *B* is generally non integer, while the minimum number of resources is integer, it is useful to also define the rounded utilization factor of algorithm *A* as

$$r(A) = \frac{\lceil B \rceil}{M(A)}, \tag{27}$$

which compares the minimum number of resources required by algorithm A to the integer lower bound. In (27), symbol $\lceil \cdot \rceil$ indicates the ceiling operator.

A drawback of the measures defined in Eqs. (26) and (27) is that they often severely under-estimate scheduling efficiency, since the lower bound B may be much smaller than the optimal number of processors required for achieving no task overlapping. Ideally, the scheduling performance of an algorithm should be compared with the optimal case. However, due to the NP-completeness of the scheduling problem the optimal number of processors is not known. In the following, we define an alternative criterion for measuring scheduling efficiency by better approximating the performance of an optimal scheduling algorithm using the efficiency factor

$$e(A)=\frac{E(\beta^{(i)})}{M(A)}, \tag{28}$$

where $E(\beta^{(i)})$ is calculated as follows. We find the number $\beta^{(i)}(t)$ of task overlapping at any given time *t*, and then define $\beta^{(i)}$ as the maximum value of $\beta^{(i)}(t)$ over all *t*, 0≤*t*<*T*, that is

$$\beta^{(i)}=\max_{t\in[0\ T]}\beta^{(i)}(t).$$

In $\beta^{(i)}$, index *i* indicates the dependence of the variable on the scheduling instance denoted as *i*. Then, $E(\beta^{(i)})$ is obtained by averaging $\beta^{(i)}$ over all instances *i* generated. Different realizations of a scheduling problem (defined by the number of tasks and their start and finish times) result in different values of $\beta^{(i)}$, which have to be computed for each problem instance *i*, before the average value $E(\beta^{(i)})$ can be obtained. Recall that the lower bound *B* depends only on the number of tasks and on their granularity and not on the specific scheduling instance [see Eq. (25)]. Thus, even though $e(A)$ is a better bound for measuring scheduling efficiency, it requires much more effort to calculate and it can only be obtained experimentally, in contrast to *B*, which is obtained analytically from parameters *g* and *N*.

Performance Results

Experimental Setup

We implemented the Spectral Clustering Scheduling (SCS) algorithm proposed, and compared its performance to that of three other algorithms proposed in the literature.

The Greedy Algorithm: The greedy algorithm assigns tasks to computational resources, one by one, taking into account the resources' current load so that no task overlapping is encountered. When a new task overlaps in time on all resources with some already assigned tasks, the new task is assigned to another resource. We implemented two versions of the Greedy algorithm, depending on the specific resource selection policy. In the first version, a resource is randomly chosen provided that no task overlapping is encountered. In the second version, a task is assigned to a resource using the following procedure. Initially, we find the resources with already assigned tasks that do not overlap with the current task. Next, among these resources, the task is assigned to the one that maximizes (after the task assignment) the minimum of the "unreserved time intervals". By "unreserved time intervals" we mean the time intervals on a resource where no tasks have been scheduled. We refer to the first approach as the Greedy Algorithm-Heuristic 1, and to the second approach as the Greedy Algorithm-Heuristic 2.

In both versions of the greedy algorithm, when a newly arrived task overlaps with some already assigned tasks on all the examined resources, the task is allocated to a new resource. The number of resources used after scheduling all tasks is denoted by *M(Greedy)*.

The Min Cut Tree Algorithm: Another scheduling scheme, we implemented for comparison purposes is the min cut tree algorithm [Leahy93] [Giles00], often used for graph clustering. In this approach, the scheduling problem is represented as a graph whose nodes correspond to the *N* tasks, and whose edges are determined by the non-overlapping degrees σ_{ij} defined in Eq. (1) (in particular, *Σ* is the incidence matrix of that graph). The constructed graph is then divided into two clusters by the application of a minimum cut technique. The minimum cut obtained through the use of a maximum flow algorithm [Fulkerson62] then corresponds to a 2 cluster partitioning. This procedure is iteratively applied so at to obtain a *k* cluster partitioning. The number of resources used after scheduling all tasks is denoted by *M(MinCut)*.

The Spectral Clustering Scheduling (SCS) Algorithm: The last scheduling algorithm evaluated is the proposed Spectral Clustering Scheduling (SCS) algorithm, which tries to jointly optimize resource utilization efficiency and degree of task requirements violations. The proposed method is recursively applied assuming different number of computational resources in the Grid. Then, we select the minimum number *M(SCS)* of resources that provide no task overlapping (that is, no violation of the tasks' requirements).

In all experiments, we assume that the tasks' start times ST_i are uniformly distributed within the time horizon *T*. If a task starts its execution within time interval *T* but its respective completion time is outside *T*, we assign the remaining time to the beginning of the interval.

Regarding the task durations $d_i, i=1,2,\cdots,N$, we consider a symmetric and an asymmetric case. In the first case, the duration of all tasks is taken to be constant and equal to *D*. In the second case, we generate tasks with high variability in their durations (very short and very long tasks), and a given average duration *D*. All results are obtained by averaging over 500 different realizations (instances) of the arrived tasks and their start times.

Simulation Results for the Symmetric Case

In this subsection, we present the results for the symmetric case, where all task durations are equal to *D*. Figure 6 presents the utilization *ρ* achieved (see Eq. (26)) versus the task granularity *g* for different values of the lower bound *B,* when the proposed SCS algorithm is used. The utilization *ρ* is a measure of the minimum number of resources an algorithm requires in order to schedule all the tasks without violating their requirements in comparison with the minimum theoretical bound *B*. In this figure, we assume *B*≥1, corresponding to a rather large number of tasks that request service. We observe that for low values of *g* the utilization *ρ* achieved initially decreases as the granularity increases. However, the rate of change decreases, and the efficiency factor remains lower bounded. We also observe from Figure 6 that for granularity values *g*≥0.2 the utilization increases as *g* increases. This is because in this case the minimum number of computational resources required for achieving no task overlapping is close to the number of tasks *N* and the utilization *ρ* approaches unity (see Eq. (26)). In the same figure, we depict the utilization versus granularity curve for different values of *B*. For a given granularity, large values of *B* correspond to more tasks that have to be scheduled.

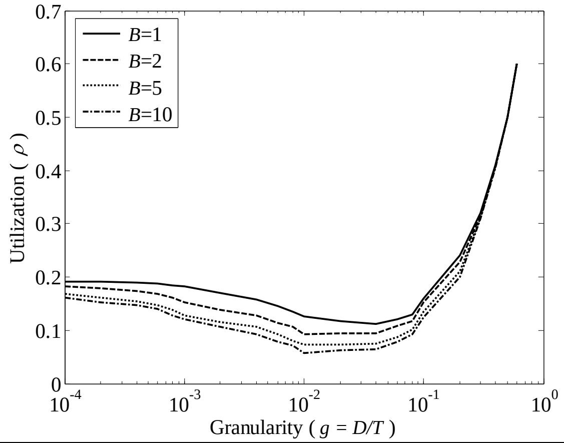


Figure 6: Utilization *ρ* versus granularity *g* for different values of the lower bound *B* when the proposed scheduling policy is used.

In Figure 7, we present the rounded utilization *r* versus granularity for values of *B* less than 1. The rounded utilization *r* compares the minimum number of resources required by algorithm *A* to the integer lower bound $\lceil B \rceil$. As expected, low values of *B* increase the scheduling efficiency. This is due to the ceiling operator involved in the definition of measure *r* (see Eq. (26)).

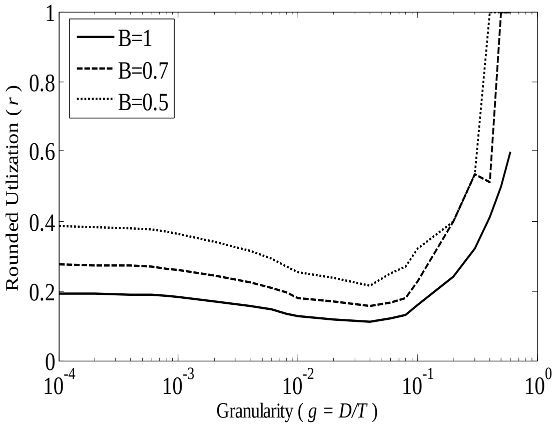


Figure 7: Rounded utilization *r* versus granularity *g* for *B*≤1.

In Figure 8, we depict the effect of the number of tasks *N* on the rounded utilization *r* for different values of the granularity *g*. We observe that the rounded utilization exhibits a periodic behaviour that depends on the granularity value. For example, for *g*=0.01 the period equals 100 task durations, while for *g*=0.04 it equals 25 task durations. This periodicity is due to the ceiling operator involved in the definition of *r* (see Eq. (26)). For large number of tasks the effect of the ceiling operator decreases, since the number of resources required also increases, and the ceiling operator then has a small effect on *r*.

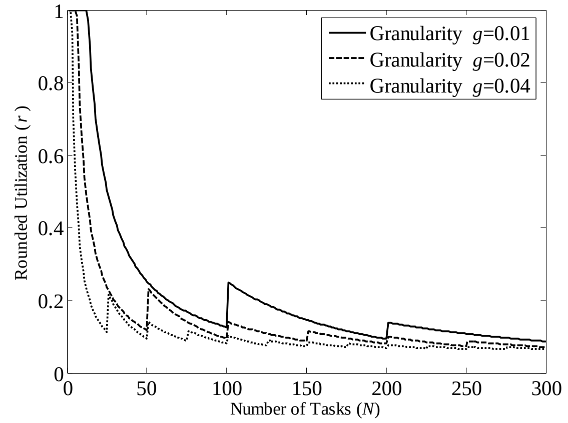


Figure 8: Rounded utilization *r* versus the number of tasks for *B*=1.

Figure 9 depicts the efficiency factor *e* versus the granularity for different values of *B*. We observe that for large values of *B* and small values of granularity the proposed scheme presents efficiency greater than 80%. In all cases the efficiency factor *e* takes very satisfactory values, indicating that on the average the SCS algorithm produces schedules that are rather close to the optimal.

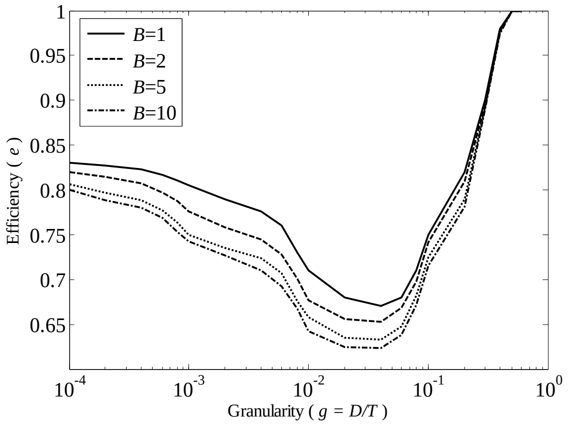


Figure 9: Efficiency *e* versus granularity *g* for different values of the lower bound *B* when the SCS scheduling policy is used.

Figure 10 presents the efficiency factor *e* versus the number of tasks. We observe that as the number of tasks increases the efficiency decreases. However, beyond a certain point, this reduction is insignificant. The efficiency factor seems to remain lower bounded and the SCS algorithm achieves no task overlap by requiring, on the average, a number of resources that is close to the optimal.

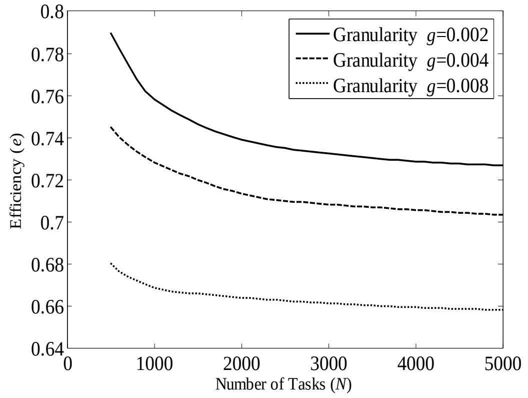


Figure 10: Efficiency *e* versus the number of tasks for different values of granularity *g* in case of *B*=1.

Simulation Results for the Asymmetric Case

In this section, we present results for the asymmetric case, where the tasks exhibit large variability in their durations *D*. The asymmetry is described as the variability margin of the task durations from their average value. A variability margin of *a*% means that the maximum and the minimum duration of the tasks vary in a uniform way ± *a*% from the average duration.

Figure 11 depicts the efficiency factor *e* versus the granularity for different values of the variability margin in the task durations. In the experiments presented, we alter the variability from 50% to 90%. As expected, when variability (asymmetry) increases, the efficiency factor decreases for low granularity values. However, for high granularity values (say, *g*>0.05) the opposite behaviour is observed, since in these cases there are several tasks of long duration compared to the time horizon *T*, making scheduling mostly unfeasible. In that case, the minimum number of resources required by the SCS algorithm is close to the lower bound $E(\beta^{(i)})$, resulting in efficiency values close to one. In order words, the application of any algorithm in these extreme cases, where tasks are long enough to occupy most of the time interval *T*, cannot yield much better performance.

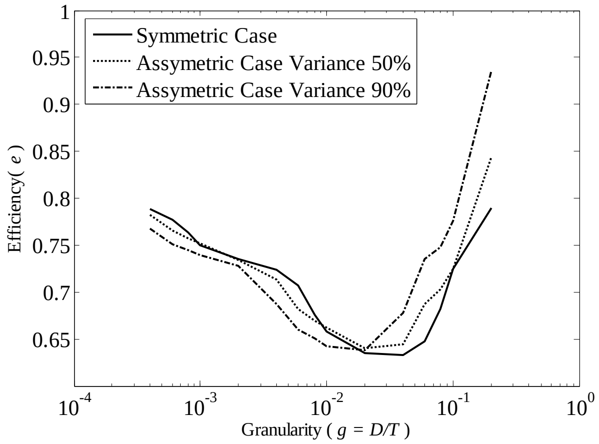


Figure 11: Efficiency *e* versus granularity for the symmetric case and different cases of asymmetric *B*=5.

Performance Comparisons between the Algorithms

We then compared the performance of the proposed Spectral Clustering Scheduling (SCS) algorithm with that of the minimum cut tree graph partitioning and of the greedy algorithms, described previously. The comparison is performed with respect to scheduling efficiency and computational complexity.

We first discuss the effect the number of iterations of the discretization method used has on scheduling efficiency. Figure 12 presents the efficiency factor of the SCS algorithm using 1, 30 and 50 iterations. The highest performance is achieved at iteration 50. However, as the number of iterations increases, the improvement benefits obtained get smaller and smaller.

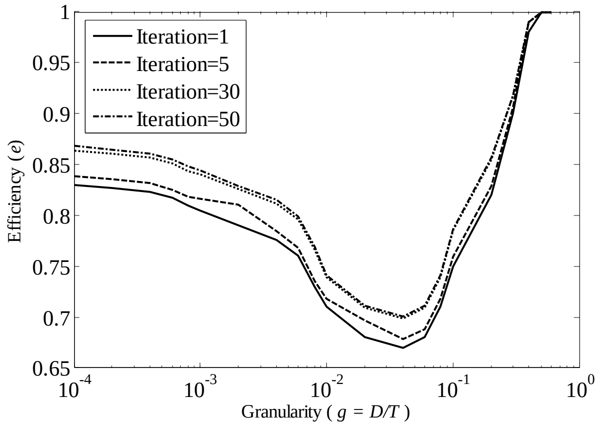


Figure 12: The efficiency-granularity curve for different number of iterations of the proposed algorithm for *B*=1.

Figure 13 compares the performance of the SCS algorithm for number of iterations equal to 1 (worst performance), to that of the minimum cut algorithm, and the two versions of the greedy algorithm. We see that (even using only a single iteration) , the SCS scheme significantly outperforms the min cut and the greedy algorithms. The simple greedy heuristic 1 algorithm presents the worst performance.

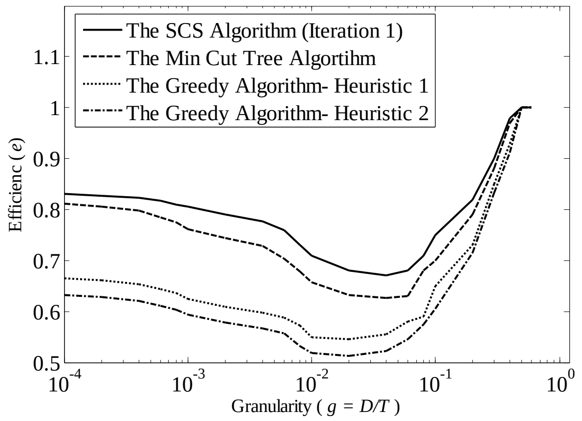


Figure 13: Comparison of the SCS algorithm with the min cut tree graph scheme and two versions of the greedy scheme for *B*=1.

2.2.2.2 *Data Consolidation*

We examine a task scheduling and data migration problem, called Data Consolidation (Figure 14). Data Consolidation (DC) applies to data-intensive applications that need several pieces of data for their execution. The DC problem consists of three interrelated sub-problems: (i) the selection of the replica of each dataset (i.e., the data repository site from which to obtain the dataset) that will be used by the task, (ii) the selection of the site where these pieces of data will be gathered and the task will be executed and (iii) the selection of the paths the datasets will follow to arrive at the data consolidating site. Furthermore, the delay required for transferring the output data files to the originating user (or to a site specified by him) should also be accounted for. Generally, a number of algorithms or policies can be used for solving these three sub-problems either separately or jointly. Moreover, the order in which these sub-problems are handled may be different from the order they are presented in this paragraph, while the performance optimization criteria used may also vary. The algorithms or policies for solving these sub-problems compromise a DC scheme.

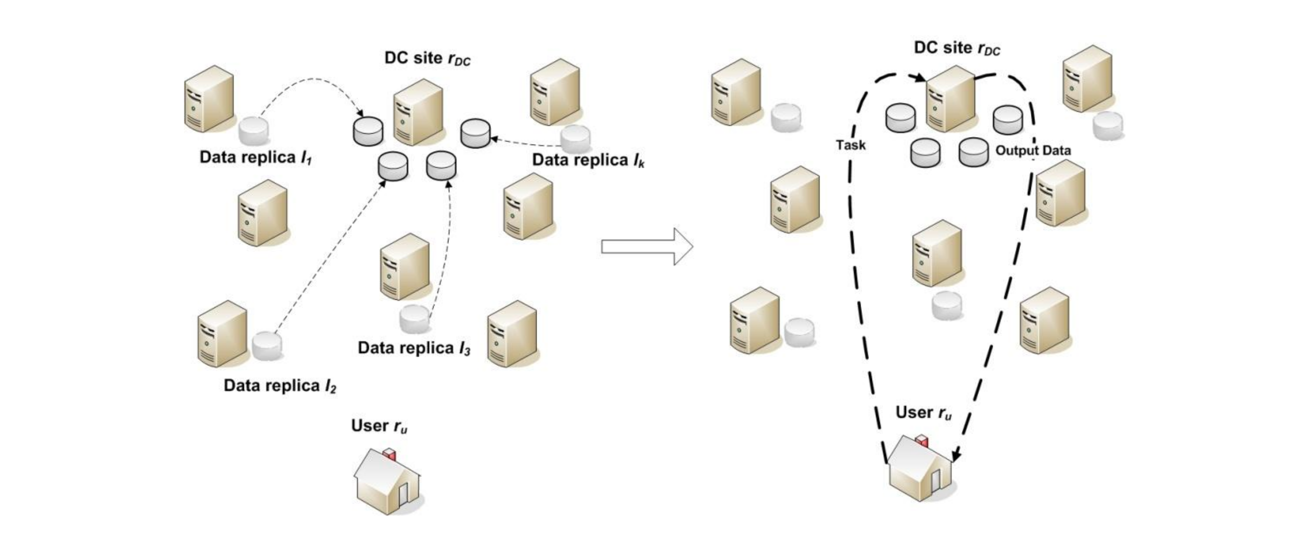


Figure 14: A Data Consolidation scenario: Initially the datasets a task requires are transferred to a single Data Consolidation site r_{DC}. After all data transfers have been completed, the task itself is also transferred to the site, where it is executed. Finally, the task's output data are transferred back to the task originating user r_u.

The DC problem appears in many Phosphorus applications and relates strongly to the placement of the storage resources and the datasets in the Grid network. We consider a number of simple DC schemes, and examine their performance by altering various parameters such as the number of data sets, the number of storage resources and the task computation and communication requirements. Generally, we show that DC is an important problem, which if performed efficiently, can lead to significant benefits in terms of task delay, network load and other performance parameters of interest.

Problem Formulation

We consider a Grid network, consisting of a set R of $N = |R|$ sites (resources) that are connected through a Wide Area Network (WAN). Each site $r \in R$ contains at least one of the following entities: a computational resource that processes submitted tasks, a storage resource where data are stored, and a network resource that performs routing operations. There are also a number of simple routers in the network. The path between two sites r_i and r_j has maximum usable capacity equal to $C_{i,j}$ and propagation delay equal to $d_{i,j}$.

The computation resource of site r_i has total computation capacity P_i, measured in computation units per second (e.g., Million Instructions Per Second - MIPS). Each resource also has a local scheduler and a queue. Tasks arriving at the resource are stored in its queue, until they are assigned by the local scheduler to an available CPU. For the sake of being specific, we assume that the local scheduler uses the First Come First Served (FCFS) policy, but other policies can also be used. At any time, a number of tasks are in the queue of resource r_i or are being executed in its CPU(s) using a space-sharing policy. The storage resource of site r_i has storage capacity S_i, measured in data units (e.g., bits). Users located somewhere in the network generate atomic (undivisible and non-preemptable) tasks with varying characteristics.

A task needs for its execution *L* pieces of data (datasets) I_k of sizes V_{I_k} , *k* =1,…,*L.* A dataset I_k has a number of replicas distributed across various storage resources. The total computation workload of the task is equal to *W*, and the final results produced have size equal to *Δ*. *W* and *Δ* may depend on the number and size of datasets the task requires. The datasets consolidate to a single site, which we will call the data consolidation site r_{DC}. The DC site may already contain some datasets so that no transferring is needed for them. The total communication delay of dataset I_k consists of the propagation, the transmission and the queuing delay. The propagation delay of path (r_i, r_{DC}) is denoted by $d_{i,DC}$ and its usable capacity by $C_{i,DC}$ (maximum capacity available at all intermediate links). A piece of data I_k transmitted over a path (r_i, r_{DC}) experiences total communication queuing delay $Q_{i,DC}^{Comm}$, because of other pieces of data utilizing the links of the path. In general the type of transport media used (opaque packet switching, transparent networks such as optical WDM network or OBS, etc), determines whether the queuing delay is counted once at the source (transparent networks) or is accumulated over all intermediate nodes (opaque networks). Finally, a task before starting execution at the DC site experiences a processing queuing delay Q_{DC}^{Proc} , due to other tasks utilizing the resource's computational capacity.

We assume that a central scheduler is responsible for the task scheduling and data management. The scheduler has complete knowledge of the static (computation and storage capacity, etc) and the dynamic (number of running and queued tasks, data stored, etc) characteristics of the sites. We do not take into account the communication delay of transferring messages between the user and the scheduler and between the scheduler and the resources, since these are negligible compared to the total execution time of the task (at least for the data-intensive scenarios that we consider in this study).

A task created by a user at site r_u, asks the central scheduler for the site where the task will execute. Upon receiving the user's request, the scheduler examines the computation and data related characteristics of the task, such as its workload, the number, the type, and the size of data needed, the sites that hold the corresponding data etc. The scheduler uses the Data Consolidation algorithm (Section 4.2) to select (i) the sites that hold the replicas of the datasets the task needs, (ii) the site where these datasets will consolidate and the task will be executed, and (iii) the routes over which to transfer these datasets. The decisions concerning (i), (ii) and (iii) can be made jointly or separately. Note that the capacity of the storage resource r_{DC} must be larger than the total size of the datasets that will consolidate:

$$S_{r_{DC}} \geq \sum_{k=1}^{L} V_{I_k} \ .$$

Next, the scheduler orders the data holding sites to transfer the datasets to the DC site. The scheduler, also, orders the user to transfer his task to the DC site (Figure 14). The tasks execution in the DC site starts only when the task and all of its needed datasets have arrived at the site. After the task finishes execution, the results return back to the originating user.

Data Consolidation

Data Consolidation Delays

We assume that the scheduler has selected the data holding sites (replicas), $r_k \in R$, for all datasets I_k, k =1,…,L, and the DC site r_{DC}. Note that the DC site may already have some pieces of data and thus no transferring is required for these pieces (i.e., r_k=r_{DC} for some k). In general, such a data-intensive task experiences both communication (D_{comm}) and processing (D_{proc}) delays. The communication delay D_{comm} of a task, considering also the delay for transferring the final results from the DC site r_{DC} to the originating user's site r_u is:

$$D_{comm} = D_{cons} + D_{output} =$$
$$\max_{k=1...L}\left(\frac{V_{I_k}}{C_{k,DC}} + Q_{k,DC}^{Comm} + d_{k,DC}\right) + \left(\frac{\Delta}{C_{DC,u}} + Q_{DC,u}^{Comm} + d_{DC,u}\right)$$

where D_{cons} is the time needed for the task's data to consolidate to the DC site r_{DC} and D_{output} is the delay of the output data to be transferred to the originating user's site r_u. The computational delay is given by:

$$D_{proc} = Q_{DC}^{Proc} + \frac{W}{P_{DC}}.$$

The total delay suffered by a task is:

$$D_{DC} = D_{comm} + D_{proc}.$$

Note that $Q_{k,DC}^{Comm}$ and Q_{DC}^{Proc} are difficult to estimate since the former depends on the utilization of the network and the latter depends on the utilization of the computation resource.

Proposed Schemes

As stated before the DC problem consists of three sub-problems: (i) the selection of the repository sites r_k from which the dataset I_k, k=1,2,…,L, will be transferred to the DC site, (ii) the selection of the DC site r_{DC} where the datasets will accumulate and the task will be executed, and (iii) the selection of the paths (r_k, r_{DC}) the datasets will follow. In general, DC schemes can make these decision based on various criteria such as the computation and storage capacity of the resources, their load, the location and the sizes of the datasets, the bandwidth availability and the expected latency, the user and application behaviours, the price a user is willing to pay for using storage and computation resources, etc.

In what follows, we propose a number of DC schemes that consider only the data consolidation (*ConsCost*) or only the computation (*ExecCost*) or both kinds (*TotalCost*, *TotalCost-Q*) of task requirements. In the proposed algorithms and in the simulation results that follow we do not take into account or simulate the time needed for the output data to be transferred to the task's originating user, assuming that it is negligible. Even though this parameter may be important in some cases, we decided to concentrate our description and our simulation results to the three more important and complex subproblems that comprise the DC problem in Data Grids, as described above.

(i) *Random-Random (Rand) algorithm:* In this algorithm the data replicas used by a task and the DC site are randomly chosen. The paths are selected using a simple Dijkstra algorithm.

(ii) *Consolidation-Cost (ConsCost) algorithm:* We select the replicas and the Data Consolidation site that minimize the data consolidation time (*D_{cons}*), assuming that the communication queuing delays are negligible.

Given a candidate DC site r_j, we select for each dataset I_k the corresponding data holding site r_i $(I_k \in r_i)$ that minimizes the transfer time:

$$\min_{r_i \in R, I_k \in r_i} \left(\frac{V_{I_k}}{C_{i,j}} + Q_{i,j}^{Comm} + d_{i,j} \right),$$

where R is the set of all resources and $d_{i,j}$ the propagation delay between site r_i and r_j. Note that in this algorithm we consider the communication queuing delays negligible and thus $Q_{i,j}^{Comm} = 0$. The data consolidation time (D_{cons}) of candidate DC site r_j, is given by:

$$D_{cons}(r_j) = \max_{k=1...L} \left(\min_{r_i \in R, I_k \in r} \left(\frac{V_{I_k}}{C_{i,j}} + Q_{i,j}^{Comm} + d_{i,j} \right) \right) \cdot$$

In ConsCost algorithm we select the DC site (r_{DC}) that minimizes the data consolidation time:

$$r_{DC} = \arg\min_{r_j \in R} \quad D_{cons}(r_j) \ \cdot$$

The paths are constructed using the Dijkstra algorithm.

(iii) *Execution-Cost (ExecCost) algorithm:* We select the DC site that minimizes the task's execution time:

$$r_{DC} = \arg\min_{r_j \in R} \left(Q_j^{Proc} + \frac{W}{P_j} \right),$$

while the data replicas are randomly chosen.

Although we cannot always calculate the processing delay Q_j^{Proc} of a resource r_j, it is possible to estimate it based on the tasks already assigned to it or based on the average delay tasks previously executed on it have experienced, etc. Moreover, if the computation workload W of a task is not known a-priori, we can simply choose the resource with the largest computation capacity P_j. The paths are constructed using the Dijkstra algorithm.

(iv) *Total-Cost (TotalCost) algorithm:* We select the replicas and the DC site that minimize the total task delay. This delay includes the time needed for transferring the datasets to the DC site, the task's execution time, and the time needed for the output data to be transferred to the task's originating user. This algorithm is the combination of the two above algorithms. The paths are constructed using the Dijkstra algorithm.

Simulation

We implemented a Data Grid Network in the Network Simulator (ns-2) [ns2]. Ns-2 provides a manageable environment for simulating the network resources of the Grid, which is particularly important for the evaluation of DC schemes. The Boost library [Boost] provided us with the implementations of the Minimum Spanning Tree algorithms.

Simulation Environment

Generally, in most data-intensive Grid applications, there is small number of advanced labs and research centers around the world, where large amounts of data (in the scale of TB and GB) are produced. These data consolidate, when needed, at a central site for processing. Furthermore, a Data Grid usually has a hierarchical structure consisting of multiple “tiers”, where each tier has its own storage capacity. Tier 0 holds all of the master files/datasets.

Our simulation environment was based on these facts. Specifically, we used the topology presented in Figure 15, which is very similar to the Phosphorus European testbed topology presented in Figure 23. Our network consists of 11 nodes and 16 links, of capacities equal to 10Gbps. In our experiments we assume a P2P (opaque) network; the delay for transmitting between two nodes includes the propagation, queuing and transmission delays at intermediate nodes. Only one transmission is possible at a time over a link, so a queue exists at every node to hold the data waiting for transmission.

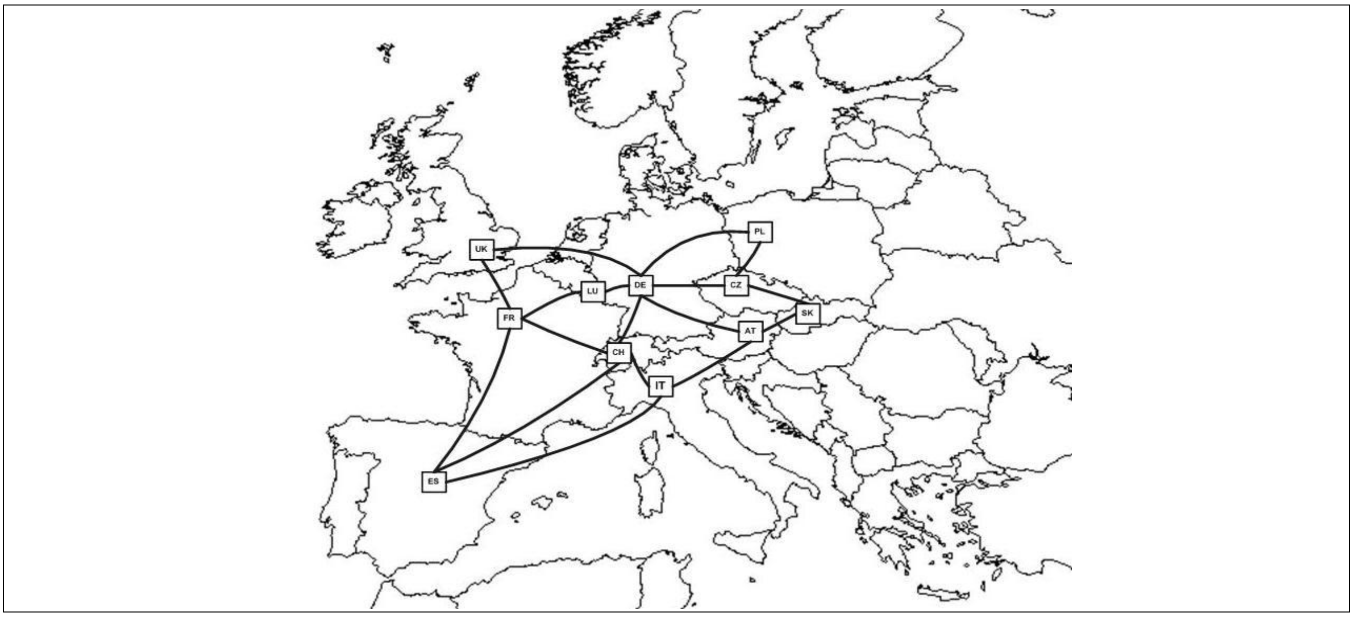


Figure 15: The topology used in our simulations.

We assume that 5 sites are equipped with computation and storage resources, while the others act as simple routers. We also assume that there is Tier 0 site in the network, which holds all the datasets, but does not have any computational capability. Each experimental scenario was run 5 times, using an independent random seed. In every repetition, the placement in the network of the 5 sites and the Tier 0 was random. Furthermore, experiments were performed using more than 5 sites equipped with computation and storage resources.

The size of each dataset is given by an exponential distribution with average *I* (GB). At the beginning of the simulation a given number of datasets are generated and two copies of each dataset are distributed in the network; the first is distributed among the sites and the second is placed at Tier 0 site. The storage capacity of each storage resource is 50% of the total size of all the datasets. Since the storage capacity is bounded, there are cases where a site does not have the capacity required to store a needed dataset. In such a case, one or more randomly chosen, unused datasets are deleted until the new dataset can be stored at the resource.

In each experiment, users generate a total of 10.000 tasks, with exponential interarrival times of average value 1/*λ*. Unless stated otherwise, we assume 1/*λ*=0.01 sec. In all our experiments we keep constant the average total data size *S* that the tasks require:

$$S = L \cdot I ,$$

where *L* is the number of datasets a task requests and *I* is the average size of each dataset. We use average total data size *S* equal to 600 GB and 800 GB and examine the (*L*, *I*) pair values presented in Table 3. In each experiment the total number of available datasets changes in order for their total size to remain the same: 15 TB and 20 TB, respectively (Table 3).

L	***I* (GB)**	**Total Number of Datasets**	**Total Size (TB)**
2	300	50	15
3	200	75	15
4	150	100	15
6	100	150	15
8	75	200	15
10	60	250	15
2	400	50	20
3	266	75	20
4	200	100	20
6	133	150	20
8	100	200	20
10	80	250	20

Table 3: The average size *I* and the number *L* of datasets each task requests. The total number of available datasets in the Grid Network and their total size is also illustrated.

The task workload *W* correlates with the average total data size *S*, through a parameter *a*, as:

$$W = a \cdot S .$$

In our simulations we use parameter *a* as follows: given the total data size *S* of a task (different for each task) and *a*, we use above equation to calculate the workload of this task. The parameter *a* measures whether a task is computation or data-intensive As *a* increases the tasks become more cpu-intensive, while as *a* decreases the tasks have less computation demands. We alter the parameter *a* and examine the performance of our DC strategies. Unless stated otherwise, in our experiments we create data-intensive tasks by setting *a* = 0.01. We also examine cpu-intensive tasks (*a* takes values up to 11). Also, when a task completes its execution we assume that there is no output data returned to the originating user.

Performance Metrics

We use the following metrics to measure the performance of the algorithms examined:

- The average task delay, which is the time that elapses between the creation of a task and the time its execution is completed at a site.
- The average load per task imposed to the network, which is the product of the size of datasets transferred and the number of hops these datasets traverse.
- The Data Consolidation (DC) probability, which is the probability that the selected DC site will not have all the datasets required by a task and as a results DC will be necessary.

The first metric characterizes the performance of the DC strategy in executing a single task, while the second expresses the overhead the DC strategy induces to the network. The third metric gives information on the way the DC site is selected, with respect to the datasets that are located (or not) at this DC site.

Simulation Results

Figure 16, Figure 17 and Figure 18 show the effects of the increased number of task requested datasets *L*, on the measured metrics and on the evaluated algorithms. Generally, given the fact that for any value of *L* the total size of data a task request is the same (equal to *S*), we believe that in order for the Grid network to better handle the case of increased *L*, it needs more sites equipped with storage resources than storage resources with increased capacity. This increases the probability that a site holding a needed dataset is close.

Figure 16 shows the DC probability for the Rand, ExecCost, ConsCost and TotalCost algorithms, when tasks request different number of datasets *L* for their execution. The higher the number *L* of datasets a task requests, the higher is the probability that these datasets will not be located at the DC site, given that the size of datasets a site can hold is limited. The ConsCost and TotalCost algorithms exhibit smaller DC probability than the Rand and ExecCost algorithms, since the former algorithms select a DC site by taking into account the consolidation delay, which is small for sites holding many or all of the datasets needed by a task. On the other hand, the Rand and ExecCost algorithms select the DC site randomly or almost randomly (as is the case for ExecCost, given that the tasks have negligible computation complexity). As *L* increases, the probability of not finding all the data at a site increases and converges to 1 for all examined algorithms.

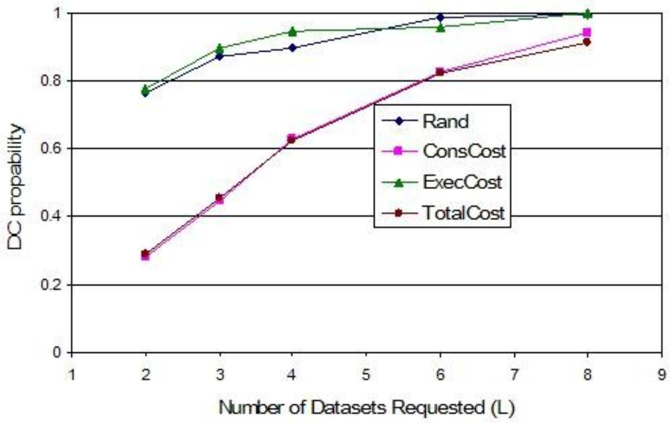


Figure 16: The DC probability for the proposed DC algorithms, when tasks request different number of datasets, *L*, for their execution. The average total data size per task is *S*=600 GB.

Figure 17 shows the average task delay for the DC algorithms examined. We observe that the algorithms that take the data consolidation delay into account (namely, the ConsCost and TotalCost algorithms) behave better than the algorithms that do not consider this parameter (that is, the Rand and ExecCost algorithms), in terms of the task delay. As the number *L* of datasets a task requires increases, the average task delays of all the algorithms converge. Specifically, for the ConsCost and TotalCost algorithms the average task delay increases as the number of datasets a task requires increases, since the probability that a DC site will not hold all the data a task needs (i.e., the DC probability) also increases (Figure 16), resulting in more data transfer. In contrast, in the Rand and ExecCost algorithms the average task delay decreases as *L* increases, because of the decrease in the size of the transferred datasets *I*. Thus, for the Rand and ExecCost algorithms that (almost) randomly select the DC site, the data consolidation time and its impact on the average task delay decreases as *L* increases.

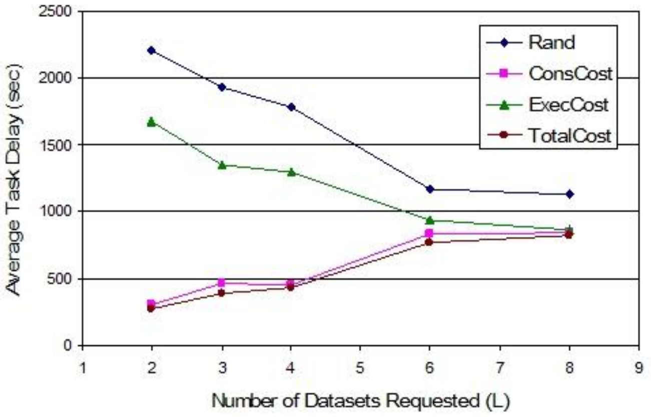


Figure 17: The average task delay (in sec) for the proposed DC algorithms, when tasks requests different number of datasets, *L*, for their execution. The average total data size per task is *S*=600 GB.

Figure 18 shows the average network load per task for the various DC algorithms, when tasks request different number of datasets *L* for their execution. We observe that the algorithms that do not take into account the data consolidation delay (that is, the Rand and ExecCost algorithms) induce, on average, a larger load on the network than the algorithms that do take this into account (ConsCost and TotalCost algorithms). This is because the former algorithms transfer on average more data, over longer paths. Moreover, the decisions made by these algorithms are not affected by the dataset sizes *I* or their number *L*, and as a result they induce on average the same network load. By analyzing our results, we observed that these algorithms transfer on average the same number of bytes over paths of equal on average length, irrespectively of *L* and *I*. The superior performance of ExecCost over that of Rand is because ExecCost assigns task to resources in a more balanced way, based on the task execution times. On the other hand, the algorithms that take into account the data consolidation delay (namely, the ConsCost and TotalCost algorithm), induce a smaller load on the network. This load increases as the number of datasets *L* increases, as can be explained by the increasing probability that a DC site will not hold all the required data (Figure 16), and will thus have to transfer more datasets.

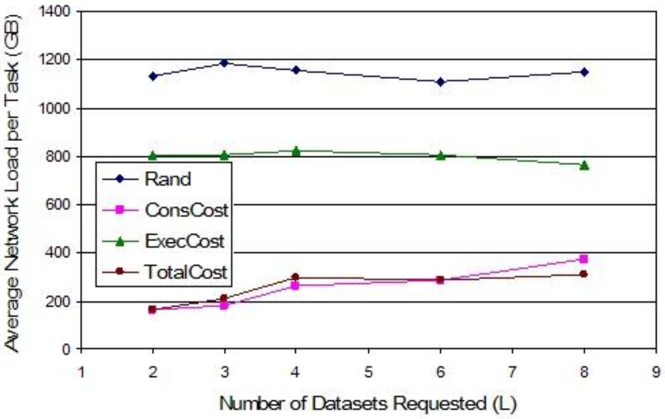


Figure 18: The average network load per task (in GB) for the proposed DC algorithms, when tasks request different number *L* of datasets for their execution. The average total data size per task is *S*=600 GB.

Figure 19 presents the results obtained when varying the number of sites, equipped with computational and storage resources, between 2 and 10, while keeping the total storage capacity of the Grid network the same as in the case when we have only 5 sites. We assume that every task request 5 datasets (*L* = 5). We observe that as the number of sites in the network increases the average task delay and the load induced to the network also increases. This is because having a larger number of sites increases the chance that the data will not be located at the DC site or at sites close to that (Figure 19c). As a result more data transfers are needed, increasing the task delay and the network load. So, the increased number of sites can also reduce the efficiency of the Grid network, in case where the number of datasets requested by the tasks is low.

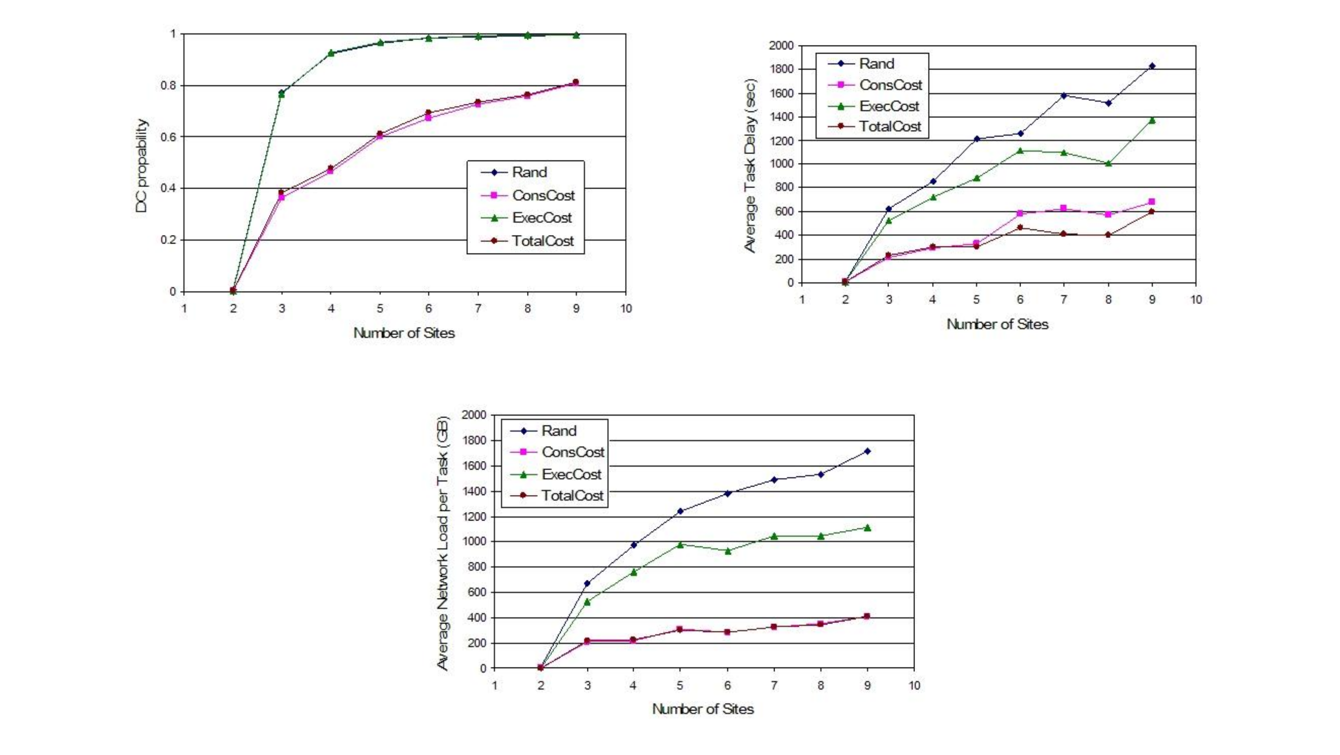


Figure 19: (a) The average task delay (in sec) and (b) the average network load per task (in GB) and (c) the DC probability for the proposed DC algorithms, when the number of sites increases. The average total data size per task is *S*=800 GB.

Finally, Figure 20 illustrates the average delay and the average network load induced per task for the proposed DC algorithms, when tasks become more cpu- rather than data-intensive. In order to examine this effect we increased the parameter *a*. We observe that the TotalCost algorithm performs better in all cases. When tasks are data-intensive, it achieves small task delay and network load and behaves similar to the ConsCost algorithm. As tasks become more cpu-intensive, the TotalCost algorithm continues to achieve small task delay and behaves similarly to the ExecCost algorithm, while the average task delay achieved by the ConsCost algorithm becomes very large. Finally, the network load induced by the TotalCost algorithm increases as tasks become more cpu-intensive, although it remains smaller than that induced by the ExecCost and Rand algorithms.

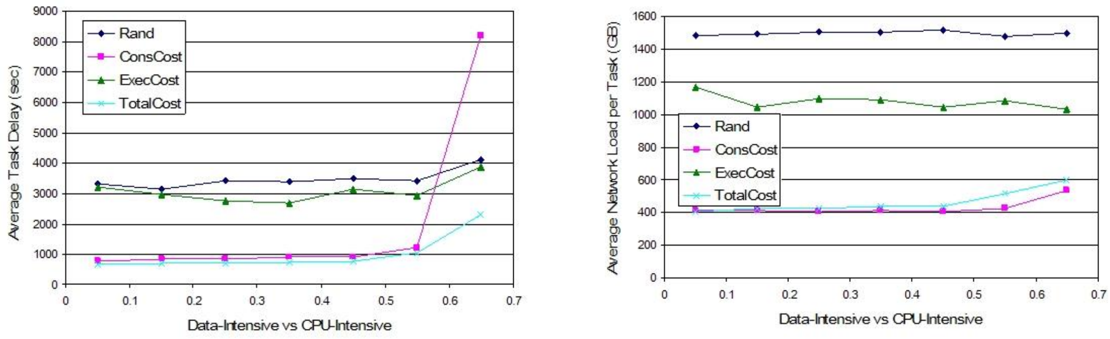


Figure 20: (a) The average task delay (in sec) and (b) the average network load per task (in GB) for the proposed DC algorithms, when tasks become more cpu- rather than data-intensive for average total data size per task *S*=800 GB.

2.2.3 Network Dimensioning

As discussed in detail in Section 1.2, the problem of network dimensioning is inherent in the process of network planning and as such constitutes a vital step in designing lambda-switched Grids as well. Despite the high volume of previous work on network design, the unique characteristics of the optical medium and their effect on dimensioning call for revisiting the problem. Quality degradation may be caused on the signal forming a lightpath due to amplifier noise, crosstalk, dispersion, filter concatenation effects and several other non-linear impairments associated with transmission of light over fibre, resulting in erroneous decoding of the signal at the receiver. These physical related issues are magnified, when the geographical reach of lightpaths is large. Considering the broad geographical scattering of a Grid network infrastructure the use of impairment-aware network design methods is inevitable [ENVI06]. Additionally, the fact that the various tasks of a Grid job may require the establishment of more than one lightpaths - in case its tasks are distributed to multiple spatially separated computational/storage resources - in conjunction with the high cost of resource usage, may make the establishment of connections in an impairment-agnostic manner impossible.

In our previous work [D.5.3], we elaborated on the impairments present in all-optical transport networks, we modelled their influence on signal quality and showed how they impact the cost of routing. The focus of this work is on investigating the benefit of deploying extra machinery that alleviates the signal degradation caused by impairments; more specifically, the employment of regenerators. 3R regenerators placed appropriately at intermediate switching points achieve to rectify the degraded signal in terms of power, shape and timing cancelling thus the effects of physical impairments. However, regeneration can be an expensive operation. Also, as currently the most mature 3R regeneration technology is based on optoelectronic conversion of the signal, carrying out regeneration introduces electronic processing, defeating the benefits of all-optical networking. For these reasons, it is desirable to minimize the number of regenerators placed in the network and also try to compare the performance of optoelectronic solutions to these of specific optical counterparts.

In most cases, the number of regenerators to use and their exact positioning in the network are issues that need to be addressed in the network design phase and therefore constitute a network dimensioning problem. In this version of the regenerator location problem [SMTFW07], [YEK03], given is only the network topology (with or without fibre links) and a set of traffic demands. The solution to the problem is then the capacity of fibre links and the location of regenerators that manages to route all demands at minimal cost (according to some cost factors). From a practical point of view, it is also worth looking at the regenerator location problem at a post-link-dimensioning phase, simply to cater for the case, where a Grid network operator decides to exploit regenerators in the growth stage, where links (and nodes) are already dimensioned.

2.2.3.1 *System Model and Node-Link Architecture*

Network Model

The detailed network model and the OXC/fibre-link architecture that have been used throughout this study are presented before delving into the details of the impairment-aware (IA) network dimensioning problem. Specifically, a WDM (Wavelength Division Multiplexed) optical network is modelled as a set of Optical Cross-connects (OXCs – referred to also as "nodes") interconnected by a set of unidirectional fibre links (referred to also as "links"). Bidirectional communication is achieved by installing two fibres between any two nodes in opposite directions. Each fibre comprises at maximum a constant number of discrete wavelengths *W* with bandwidth *B*, *W* and *B* depending on the WDM technology deployed by the lambda-switched Grid. Every OXC can be both a terminal and/or an intermediate node of a lightpath, whereas wavelength conversion capability is not an option in our model, i.e. wavelength continuity is a tight constraint. To cater for the case, where the aggregate bandwidth capacity of a link's single fibre does not suffice for serving the cumulative bandwidth demand of the lightpaths traversing the link, a link may carry more than one fibre. Note that this is straightforward, if bidirectional communication among any two nodes is a requirement. We limit the maximum number of fibres that can be installed in a duct by a small integer *maxfibre*.

Traffic and Routing Model

Since we are solving a dimensioning problem, we assume prior full knowledge of traffic requests. More precisely, we assume a two-dimensional traffic matrix *T* reflecting the estimated bandwidth demand between any two Grid sites. Each entry *T(i,j)* of the matrix specifies the number of lightpaths that need to be installed to carry the traffic generated at Grid site i and has Grid site j as its destination. Unless otherwise stated, we assume that the input traffic matrix captures the worst case among all possible simultaneous bandwidth requirements among any two nodes. Specifically, if the actual traffic demand in terms of number of wavelengths originating at node *i* and being terminated at node j takes integer values in $[a,b]$ $(a,b \in \square_{+})$, then $T(i,j) = \sup(a,b) = b$. $T(i,j) = \sup(a,b) = b$.

Furthermore, lightpaths are routed in our model using shortest paths, where path length is defined as the summation of the lengths (in kilometres) of all physical links comprising the path. Since it is evident that the effect of physical impairments is partly proportional to the topological distance covered by a lightpath, it seems unnatural to use any other sense of shortest path routing other than physical distance. Apart from the shortest path between the source and the destination of a traffic demand, we also consider alternative paths using a k-shortest path algorithm specified in [MMS07] with the same definition of path length as above. The intuition behind considering additional paths being longer than the single shortest path will be explained later in this section.

Last, throughout this work, we do not make any provision for resilience, i.e. all lightpaths installed during the dimensioning process are working paths. Since this works constitutes a first principles study of the influence of physical impairments to network design, we leave the incorporation of resilience considerations into dimensioning as future work.

Node/Link Architecture

The architecture of nodes and links considered in this section is identical to the architecture assumed in [D.5.3] for the modelling of physical impairments. For the sake of completeness, we briefly review the two models.

We assume the wavelength selective node architecture shown in Figure 21, where N_f is the number of fibres deployed at a link incident to the node ($N_f \leq maxfibre$). Each fibre of an input link carries maximally W wavelengths that are first demultiplexed. Subsequently, each wavelength λ_i is routed through the respective switch fabric and is either terminated, regenerated or routed to the appropriate output port, where all wavelengths are multiplexed and transmitted through the output fiber. Two are the implications of using this particular node architecture to the network-dimensioning approach presented here: a) it influences the analytical modelling of the power penalty introduced by Crosstalk (see section 4.1.1 in [D.5.3]) and b) it affects the modelling of a node's dimensions (number of input fibres times number of output fibres), as it will be outlined in section 2.2.3.2. Additionally, we assume that every wavelength selective switch at an OXC with *N* incident fibers is equipped with N additional ports for termination/regeneration purposes, i.e. if *N* is the number of incident fibers to the OXC, then each wavelength specific switch is of size *2Nx2N*. This is a realistic assumption in the context of Grid optical networks, since every OXC acts as a point-of-presence for a Grid site and therefore every OXC is a candidate for connection initiation/termination. In the case of core networks, where the highest fraction of incoming traffic is switched and not terminated, the validity of the above assumption depends clearly on the cost per input port relative to the rest of the cost factors contributing to the total capital cost of the network. Thus, even in core networks, if the cost per input port for the switch dimensions considered is negligible compared to the rest of the factors contributing to total cost, the choice of adding N redundant ports does not lead to large deviations from optimal cost.

Figure 22 depicts the link architecture we consider, which is thoroughly outlined in section 5.3 of [D.5.3]. Particular to the work reported in this document, it is additionally noted that the cost of a fibre link is clearly proportional to the link length, due to the fact that the required machinery (amplifiers, dispersion compensation units etc.) is installed per span and the span length is constant. The parameter values used for the various fibre types (SMF, DCF) are listed in Appendix A.1.

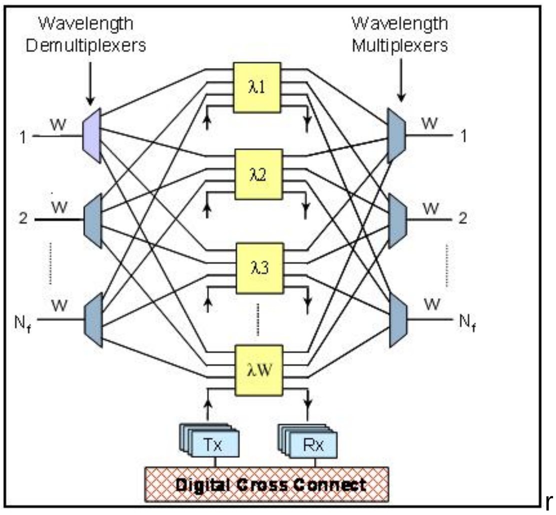


Figure 21: Node model assumed throughout impairment-aware network dimensioning

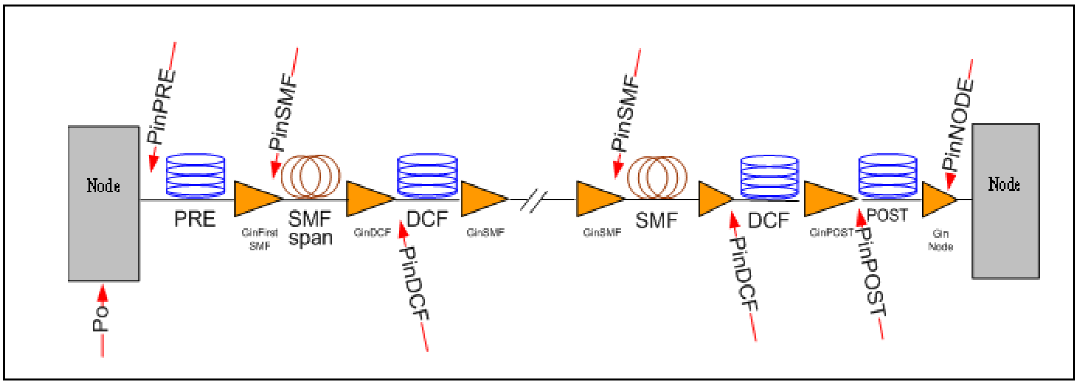


Figure 22: Architecture of a fibre link assumed in our model (single fibre shown)

2.2.3.2 *The Case for IA-Network Dimensioning*

A lambda-switched infrastructure used to interconnect Grid sites has to guarantee acceptable signal quality for the lightpaths provisioned to serve the Grid's traffic demands. Otherwise, communication among pairs of sites will be either totally impossible or of unacceptable quality, leading to low utilization of resources. In this subsection, we investigate the effectiveness of traditional network dimensioning in providing lightpaths with acceptable quality of decoded signal, assuming a transparent WDM network (no regeneration). Since physical impairments of WDM networks are evidently magnified in networks spanning large geographical areas, the type of investigation we propose proves to be necessary prior to designing the Phosphorus optical network infrastructure, which spans a large fraction of the entire European continent.

Physical Impairments

Physical layer impairments may be classified as linear and non-linear. Linear impairments are independent of the signal power and affect each of the optical channels individually, whereas nonlinear impairments are signal power dependent and affect not only each optical channel independently, but also cause disturbance and interference between them. The following physical impairments are considered in this analysis:

- **Amplified Spontaneous Emission (ASE)** noise: noise introduced by amplifiers deployed at the end of each fiber span (linear impairment).
- **Chromatic dispersion:** quality degradation due to broadening of the signal as it travels through the fiber (linear impairment).
- **Self-Phase Modulation (SPM):** quality degradation due the dependence of the optical phase of a wavelength on its own intensity (non-linear impairment).
- **Cross-Phase Modulation (XPM):** impairment due to non-linear phase shift of the signal carried by a wavelength due to the intensity of neighboring channels (non-linear impairment).
- **Four Wave Mixing (FWM)**: causes degradation of the optical signal due to interference with unwanted optical components generated by the nonlinear interaction of optical channels propagating in the same fiber (non-linear impairment).

We analytically modelled the degradation in signal quality caused by each of the above impairments in [D.5.3]. In the present deliverable, we apply these analytical models to quantify the signal degradation in lightpaths installed during the dimensioning phase. The values of the various parameters throughout the impairment models can be found in [D.5.3].

The metric we use to quantify the quality of the signal carried by a lightpath *p* is the signal Bit Error Rate (*BER*) at the receiver. The *BER* of a lightpath *p* is related to the Q-factor as follows:

$$BER(Q_p) = \frac{1}{2} \cdot erfc(\frac{Q_p}{\sqrt{2}})$$

The Q-factor Q_p is defined as the electrical signal-to-noise ratio at the input of the decision circuit of the receiver of lightpath *p*. Particular to the impairments considered herein, the Q-factor of a lightpath *p* is calculated according to the following term:

$$Q_p = pen \cdot \frac{P}{\sqrt{\sigma_p{}^2}}$$

where Q_{SPM} is the Q-factor caused by SPM/GVD phenomena, *P* is the launch power and σ_P is given by:

$$\sigma_p^2 = \sigma_{ASE,i}^2 + \sigma_{XPM,i}^2 + \sigma_{FWM,i}^2$$

with $\sigma^2_{XPM,i}$ and $\sigma^2_{FWM,i}$ being the electrical variances of the XPM and FWM induced degradations respectively and $\sigma^2_{ASE,i}$ being the electrical variance of the ASE noise, all measured for lightpath *p*. Note that the effects of chromatic dispersion (CD) are considered through the analytical equations of each non-linear impairment

Cost Model

Given a topology of Grid sites and a set *E* of candidate physical links interconnecting the sites, the goal of network dimensioning is to specify a subset of *E* and the capacity –number of fibres and wavelengths per fibre – installed on each physical link, such that all demands of the input traffic matrix are served. Additionally, among all possible link/capacity configurations that serve the input traffic demand, the one with minimum cost has to be specified. For this, an accurate and realistic model quantifying the costs associated with deploying a physical link and adding fibre/wavelength capacity to it is required. We adopt the linear model presented in [VVDD98], however we introduce some amendments to it to make the model more realistic and suitable for the specific dimensioning requirements of this study. Throughout the analysis, we differentiate and define costs for two distinct instances of the network dimensioning problem, both of practical interest:

a) **"Greenfield" Network Dimensioning (G-ND)**: in this instance of the problem, only the geographical location of Grid sites is given and a set of candidate links for interconnecting them, which however do not physically exist, i.e. trenching is required for each link that the dimensioning outcome mandates installing between two sites.

b) **"Deployed" Network Dimensioning (D-ND)**: unlike the previous instance of the problem, ducts are already in place and equipped with a number of fibres (constant number – uniform across all links). Thus, the entire topology (nodes and links) is given and the role of dimensioning is to specify the number of fibres per link and wavelengths per fibre that need to be activated to serve the input traffic demand.

We start with modelling the cost of installing the cable link between two Grid sites during G-ND, which involves the cost of digging the duct, leasing cost, fibre cost and maintenance cost. Intuitively, the cost of this process is proportional to the geographical distance covered by the link, since not only the installed hardware (pipes) and leasing cost is proportional to link length, but also the cost associated with the time required to complete the trenching (labour cost, equipment leasing) are proportional to link length. Let l_i be the length of link *i* and α stand for the trenching cost per link length unit (e.g. monetary unit/km). Then, the trenching cost α_i for link *i* is given by $\alpha_i = \alpha \cdot l_i$. In the scope of the D-ND problem, only the leasing and maintenance costs should be considered.

Unlike trenching cost, fibre cost in the G-ND problem comprises two components: one proportional to link length and an additional fix cost per installed fibre. In the scope of the G-ND problem, the length dependent cost is due to machinery installed per fibre span, like amplifiers, dispersion compensation fibre and pre-/post-compensation fibre (see subsection 2.2.3.1). The fixed cost component accounts for equipment installed at the termination points of the fibre, like de-/multiplexers. Let β_{span} and l_{span} l_{span}stand for the cost per span and the span length respectively and $\beta_{fix}$$\beta_{fix}$ stand for the fix cost of installing a fibre (all considered constant – uniform to all fibres). Then, the total cost β_i of installing a fibre on link *i* of length l_i is:

$$\beta_i = \frac{l_i}{l_{span}} \cdot \beta_{span} + \beta_{fix}$$

The same cost model for a fibre link applies also to the D-ND problem, since we assume only the existence of dark fibre at each link prior to dimensioning, i.e. all termination and per span equipment will have to be installed in this case too, when a fibre is lit up.

In both problem instances considered – G-ND and D-ND – the sole cost of installing/activating a wavelength is the cost *γ* of the transmitter/receiver associated equipment, which is constant per wavelength. Thus, if γ_{ij} denotes the cost of installing/activating wavelength *j* on link *i*, then $\gamma_{ij}=\gamma$.

Despite the fact that the dimensioning procedure deals only with links and link capacities, the total cost it incurs is not only influenced by the selection of link set and capacities to be added, but also by the selection of the switch fabric dimensions at the OXCs. This is particularly true in both problem instances. For the node architecture assumed herein (see subsection 2.2.3.1), two are the factors that drive the cost of the switch fabric at a node: a) the number of MEMS switches required, which equals the maximum number of wavelengths deployed among all fibres incident to the node and b) the dimensions of each wavelength selective switch, which depends on the number of fibres incident to the node. For the sake of keeping the problem formulation in the linear domain, our model assumes that the cost of switching is only governed by factor b), namely the dimensions of each of the wavelength selective switches and is in fact independent from the cumulative number of switches at each node. Differently put, we assume that all nodes are equipped with W switches, one for each of the wavelengths that could possibly be switched via the node. We assert that this simplifying assumption does not lead to a high deviation from the optimal minimum of the network's cost, since - across all nodes - at least one fibre incident to a node will be with high probability equipped with a number of wavelengths that approaches W. Based on the above, we consider *S* distinct sizes of rectangular switches (e.g. 4x4, 8x8 etc.) and assign a constant cost per input port φ_s ($s=1..S$) to each switch of type s.

Quantifying the Effect of Physical Impairments

To showcase the need for incorporating impairment-awareness in the network dimensioning process, we solve the G-ND and D-ND problems using the two topologies proposed for interconnecting Grid sites in Phosphorus.

Towards this end, we first formulate the G-ND problem as an Integer Linear Program (ILP). Let $G = (V,E)$ be the graph corresponding to the input topology, where *V* is the set of nodes (Grid sites) and *E* is the set of candidate links. Let also $f : E \rightarrow Q$ be the transformation that returns for each link the distance l_e between the two nodes that it is incident to. Given is also a $|V| \times |V|$ traffic matrix *T* with zero diagonal as specified in section 2.2.3.1. In the following, instead of using index pairs to refer to elements of the incidence matrix of G and to elements of *T*, we instead serialize access by using integer indices. More precisely, we refer to links using the integer index $e(1 \leq e \leq |E|)$ and to traffic demands using the integer index $d(1 \leq d \leq D)$, *D* being the number of non-zero elements of T; thus traffic demands requesting no lightpaths are ignored. Furthermore, we use the integer index $c(1 \leq c \leq W)$ to refer to each of the distinct wavelengths installed on a fibre. Let also $h_d\ (d = 1..D)$ be initialized to the number of wavelengths requested by demand *d*. Towards modelling the dimensions of a node's switching fabric, we define two more constants: a) $\eta_{e,n} \in \{0,1\} (e = 1..|E|, n = 1..|V|)$,which is set to 1, if link *e* is incident to node *n,* and is zero otherwise and b) $\theta_s \in \square^{*}_{+} (s = 1..S)$ is an integer corresponding to one of the dimensions of an *nxn* switch (e.g. $\theta_s=4$ for a *4x4* switch), *S* being the number of possible switch dimensions considered by the problem. For the problem instances considered in this work, we allow θ_s to take values in the set {2, 4, 8, 16}, i.e. we set:

$$\theta_1 = 2 \ (2x2 \ switch)$$
$$\theta_2 = 4 \ (4x4 \ switch)$$
$$\theta_3 = 8 \ (8x8 \ switch)$$
$$\theta_4 = 16 \ (16x16 \ switch)$$

As already noted, we consider shortest-path routing solely for routing lightpaths, using geographical distance as the link weight metric. In fact, we consider the k shortest alternative paths between source and destination of a demand (k being a small positive integer) and allow the ILP to decide which of these k paths economizes the most in the total network cost. For this, we run for each demand *d* the k-shortest path algorithm with graph G and the label function ƒ*f* as its input. We capture the output of the algorithm in a binary constant δ_{edp}:

$$\delta_{edp} = \begin{cases} 1, \ if \ \ link \ e \ is \ used \ by \ path \ p \ to \ serve \ demand \ d \\ 0, otherwise \end{cases} \quad , e = 1..|E|, d = 1..D, p = 1..k$$

The integer variables, the optimal values of which the ILP seeks to specify, are as follows:

- $x_{dpc} \in \mathbb{Z}_+$: Number of lightpaths that use the *pth* path serving demand *d* on wavelength *c*.
- $w_{ce} \in \mathbb{Z}_+$: Number of times wavelength *c* is used on link *e*.
- $y_e \in \mathbb{Z}_+$: Number of fibers installed on link *e*.
- $u_e \in \{0,1\} = \begin{cases} 1, \ if \ \ link \ e \ is \ used \ in \ the \ \mathrm{dim}ensioned \ network \\ 0, otherwise \end{cases}$
- $t_{n,s} \in \{0,1\} = \begin{cases} 1, \ if \ \ node \ n \ requires \ switch \ fabric \ \mathrm{dim}ensions \ corresponding \ to \ \ switch \ type \ s \\ 0, otherwise \end{cases} \quad , n = 1..|V|, s = 1..S$

The objective of the optimization problem is to minimize the total cost of the dimensioned network. According to our cost model, the total cost *Z* of a dimensioned network is given by:

$$Z = \sum_{e=1}^{|E|} (u_e \cdot \alpha_e) + \sum_{e=1}^{|E|} (y_e \cdot \beta_e) + \sum_{e=1}^{|E|} \sum_{c=1}^{W} (w_{ce} \cdot \gamma) + \sum_{n=1}^{|V|} \sum_{s=1}^{S} (t_{n,s} \cdot \varphi_s \cdot \theta_s \cdot W)$$

In the above objective function, the cost per input port φ_s of a single selective switch is multiplied by the number of ports θ_s of the switch, which yields the cost of switching a single wavelength at node n. The total switching cost of the node is obtained by multiplying this product by the number of wavelengths switched at node n (set equal to W as explained previously).

The first constraint on the values that the variables of the ILP may take mandates that all traffic demands should be served, therefore the following must hold:

$$\sum_{p=1}^{K}\sum_{c=1}^{W} x_{dpc} \geq h_d \ ,d = 1..D$$

Moreover, the following two capacity constraints must hold:

$$y_e \geq w_{ce} \ ,c = 1..W ,e = 1..|E| \quad \text{and}$$

$$\sum_{d=1}^{D}\sum_{p=1}^{K}(\delta_{edp} \cdot x_{dpc}) \leq w_{ce} ,c = 1..W ,e = 1..|E|$$

The first of the above terms forces the ILP to equip a link with a number of fibres that is at least equal to the number of times each distinct wavelength uses the link. The second term mandates that (at each link) a distinct wavelength must occur a number of times that is equal to the number of lightpaths using this particular link-wavelength combination.

The maximum number of fibres that can be installed on a link – in case the link is used in the dimensioned network – is upper-bounded by the *maxfibre* constant:

$$y_e \leq u_e \cdot max\,fibre \ ,e = 1..|E|$$

Last, we add two more constraints that govern the values that a node's switching fabric dimensions could take during the optimization process. The first constraint essentially states that each node's (rectangular) switching dimensions can take only one value among all *S* possible dimensions:

$$\sum_{s=1}^{S} t_{n,s} = 1 \, , n = 1..|V|$$

whereas the second constraint specifies that the dimensions of a node's switching fabric should be sufficiently high to switch among the number of fibres incident to the node:

$$\sum_{e=1}^{|E|}(2 \cdot \eta_{e,n} \cdot y_e) \leq \sum_{s=1}^{S} \theta_s \cdot t_{n,s} \ ,n = 1..|V|$$

The multiplicative factor of 2 in the last term is added due to the fact that we require for a node with N incident fibres to be a equipped with 2Nx2N switch; this caters for the worst-case of N connections being terminated (drop) and N connections being initiated (add) at the node. The integer program formulated above uses $D \cdot k \cdot W + W \cdot |E| + 2 \cdot |E| + |V| \cdot S$ variables and $D + 2 \cdot W \cdot |E| + |E| + 2 \cdot |V|$ constraints. The formulation can be found in compact form in Appendix 6B.1.

The ILP formulation of the D-ND problem follows easily from the formulation of the G-ND by removing from the objective function the cost sum accounting for trenching. The objective function Z' of the D-ND thus becomes:

$$Z' = \sum_{e=1}^{|E|}(y_e \cdot \beta_e) + \sum_{e=1}^{|E|}\sum_{c=1}^{W}(w_{ce} \cdot \gamma) + \sum_{n=1}^{|V|}\sum_{s=1}^{S}(t_{n,s} \cdot \varphi_s \cdot \theta_s \cdot W)$$

Otherwise, the ILP formulation of the D-ND is identical to that of the G-ND.

We implemented both ILP formulations in Matlab 7.0 and used the GNU Linear Programming Kit (GLPK) open-source solver to obtain optimal solutions to the various instances of the problem. Figure 23 illustrates the topology assumed for the Phosphorus testbed, which constitutes also the input topology used to create instances of the dimensioning problems.

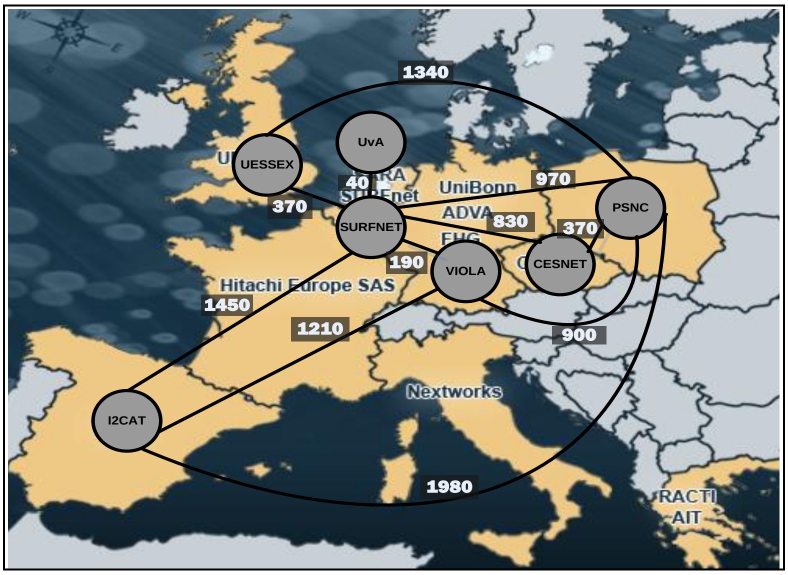


Figure 23: Phosphorus European testbed topology (link labels correspond to link lengths in km)

Table 4 lists the values used for the various input parameters to instantiate an instance of the ILP problem. Each parameter set forms an instance of an ILP, whose solution reveals the set of links and the fiber/wavelength capacity on each of the links such, that the total cost of the network is minimized. Additionally, the solution to the ILP maps univocally a lightpath to each input traffic demand.

Parameter Name	*G-ND Problem*	*D-ND Problem*
Topology	a) Phosphorus European Testbed b) Phosphorus Global Testbed	a) Phosphorus European Testbed b) Phosphorus Global Testbed
Number of nodes	7	9
Number of links	22	28
#Demands (cumulative λs)	200	200
#demands per src/dst pair	Uniformly distributed in [2,6]	Uniformly distributed in [2,6]
#Alternative shortest paths considered	k =3	k=3
#Maximum λs per fiber (W)	40	40
Channel spacing	50 GHz	50 GHz
#Maximum Fibers per link	Unlimited	10
Span length l_{span}	80km	80km
# Switch Sizes considered	S=4	S=4
Dimensions of Switch Types	$\theta_1 = 2, \theta_2 = 4, \theta_3 = 8, \theta_4 = 16$	$\theta_1 = 2, \theta_2 = 4, \theta_3 = 8, \theta_4 = 16$
Maximum BER threshold	$< 10^{-15}$	$< 10^{-15}$
Cost γ per wavelength	1	1
Cost per span relative to cost per wavelength	β_{span}=1	β_{span}=1
Fiber termination cost relative to cost per wavelength	β_{fix}=1.25	β_{fix}=1.25
Trenching Cost per km relative to cost per wavelength	a=1	a=0.1

Cost per input port of switch of type *s* (s=1..S) relative to cost per wavelength	s=1: φ_1=0.05 (2x2 switch)
	s=2: φ_2=0.2 (4x4 switch)
	s=3: φ_2=0.5 (8x8 switch)
	s=4: φ_3=1.5 (16x16 switch)
		s=1: φ_1=0.05 (2x2 switch)
		s=2: φ_2=0.2 (4x4 switch)
		s=3: φ_2=0.5 (8x8 switch)
		s=4: φ_3=1.5 (16x16 switch)

Table 4 - Input parameter values used for the generation of the dimensioning problem instances solved

To rate the necessity for impairment-aware network dimensioning, the Bit Error Rate (BER) of each lightpath is calculated at its termination OXC using the analytical model described earlier in this subsection. A lightpath is considered impaired, if its calculated BER exceeds the assumed threshold. We first elaborate in the optimal solution obtained for the "greenfield" problem instance, using the Phosphorus European Topology as input. Additionally, we experiment with two value sets for the parameters modeling the noise figures of the amplifiers deployed across each physical link. All noise figures are set equal in the first set (termed "homogeneous network"), whereas the noise figures in the second set (termed "heterogeneous network") are randomly selected from a pool of realistic noise values.

Figure 24 shows the scattering of BER values against path length for all lightpaths created to serve the traffic demands (note that routes with the same length and BER are consolidated into a single dot). We first observe that in the homogenous scenario there is an interval of path length values that clearly separates the space to impaired and non-impaired paths. This indicates that using path length may be a reliable criterion for placing regenerators in this particular scenario. However, this is not true in the heterogeneous case, where BER is not necessarily increasing with increasing path length; instead, and due to the heterogeneity built into the network, signal quality on a lightpath is not consistently proportional to path length. Therefore, deciding where to place a regenerator based on an assumed optical reach is not clear to be an effective strategy in the heterogeneous case.

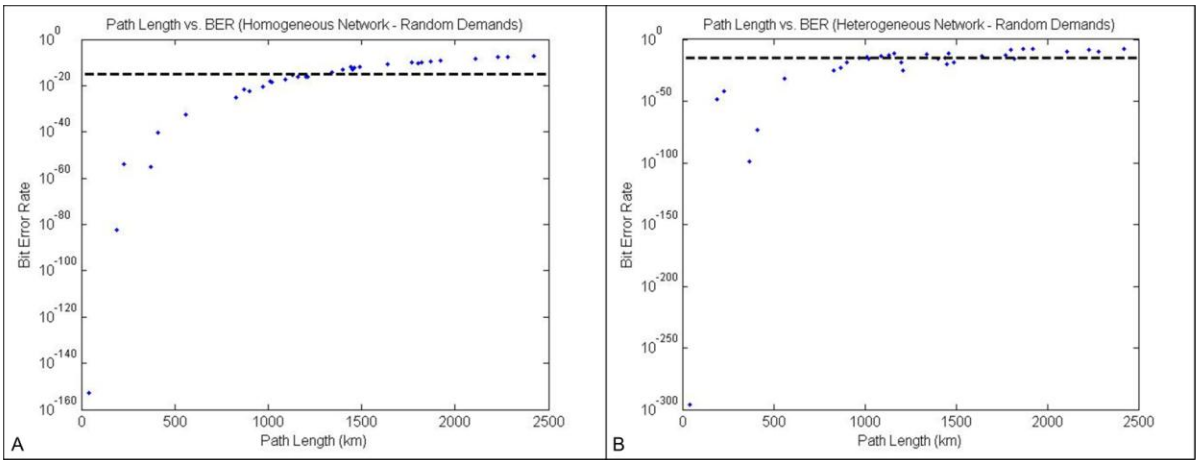


Figure 24: Bit Error Rate of installed lightpaths against physical length in a dimensioned homogeneous (A) and heterogeneous (B) network. Note that each point in the space may and in fact does account for multiple lightpaths exhibiting same length and BER.

To quantify the effect of placing regenerators according to path length, the following experiment was conducted: using the previously described parameter value set and for the heterogeneous network case, the optimal solution to the G-ND problem is first found; subsequently, length-based regenerator placement is applied. More precisely, for each node *i* of an installed lightpath *p* in the dimensioned network, a regenerator is placed at node *i*, if the physical distance from the last regeneration point to the next node of *i* on lightpath *p* exceeds the optical reach *L.* Note that the source node of the demand served by lightpath *p* is considered as a regeneration point. We experimented with values of *L* in the [500,2500] kilometre range, stepping by 10km between two consecutive optical reach values.

Figure 25 depicts the trade-off between number of redundant regeneration points and percentage of impaired paths for the various optical reach values. The most obvious observation that emanates from these results is that typical values of assumed optical reach (usually *L* > 1500km is assumed) yields unacceptably high percentage of impaired paths; this contradicts the fundamental network design requirement of serving 100% of the input demands. In our experimental setting, the requirement of serving all demands – which is equivalent to having no impaired paths – is satisfied for optical reach values that are lower than 980km. Still, this comes at the expense of regenerating lightpaths that exhibit acceptable signal quality, despite exceeding the optical reach value: 40 redundant regenerators are required at minimum, which accounts for 52% of the total regeneration capacity installed into the network (over 100% redundancy). The results were similar in the D-ND problem solution.

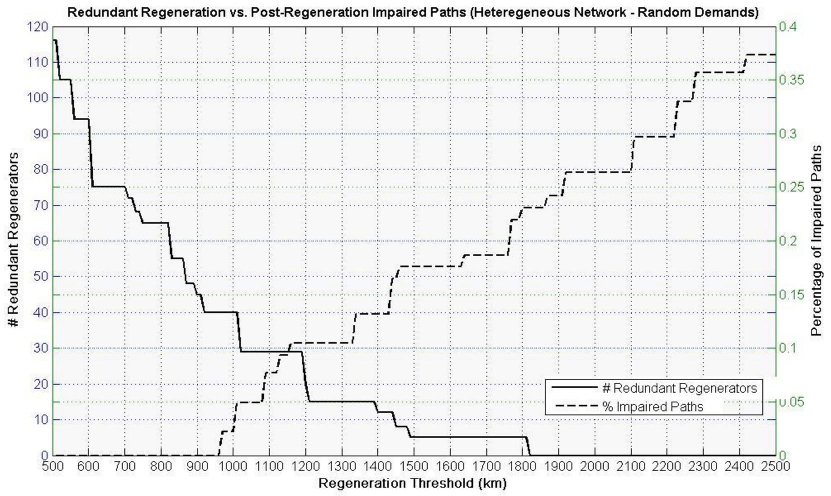


Figure 25: Trade-off between fraction of impaired paths and redundant regeneration for length-based regeneration in a heterogeneous configuration of the EU-Phosphorus testbed topology.

Note that the lack of configuration data from deployed optical networks did not allow us to specify the frequency, at which such large redundancy costs occur. Still, configuring heterogeneity in our experimental setting with realistic values is a clear indication that such cases can occur. The present work is a first-principles approach to impairment-aware network design and as such specifying the frequency of certain levels of heterogeneity is part of our ongoing work.

2.2.3.3 *Motivation and Problem Statement*

The results presented in the previous subsection clearly indicate that the use of length-based regeneration is suboptimal in optical networks with high degree of optical equipment heterogeneity. Since acceptable signal quality for all demands is a requirement during the network design phase, it follows that this approach is plagued with redundant equipment costs. Conversely, if a high regeneration threshold is used – as can be safely assumed in homogeneous networks – the cost of rectifying the impaired paths will appear at the network deployment phase.

This shortcoming of designing translucent networks motivates more involved network dimensioning methods that guarantee deterministically acceptable signal quality for all lightpaths and that do not come with hidden costs that will arise during later phases of network evolution. Since network dimensioning should yield minimum cost network designs independent of the diversity of technologies used, this implies that advanced dimensioning methods ought to incorporate signal quality awareness into the problem. In the following, the already presented dimensioning methods are enhanced with additional optical constraints to address all the above issues. The new approaches are evaluated via simulation in the Phosphorus testbed topologies and the benefit against traditional network dimensioning is shown.

2.2.3.4 *Impairment-aware Network Dimensioning in Phosphorus*

Our advanced network dimensioning method builds on the already presented formulations of the G-ND and D-ND problems as mathematical programs. In fact, the impairment-aware versions of these two ILPs are identical to their basic counterparts with regard to constraints, whereas they slightly differ in the objective function formulation and in the pre-processing steps of input constants. Throughout this work we consider optoelectronic regeneration only (OEO regeneration) and therefore we refer to it hereafter simply with the term “regeneration”. Also, wherever we refer to evaluation of signal quality of a path (or sub-path), a comparison between the calculated Bit Error Rate (BER) across the (sub-)path and a given BER threshold is implied; a path’s BER is computed using the analytical model outlined in subsection 2.2.3.2.

As pointed out during the specification of the G-ND ILP formulation, the input paths that are candidate for serving a demand are the *k*-shortest among the source and the destination of the demand. In the **IA-G-ND (“Impairment-Aware Greenfield Network Dimensioning”) problem**, additionally to finding the *k*-shortest paths, the least number of regeneration nodes required to rectify each of the *k* paths are computed. It is straightforward that this only applies to paths that are found to be impaired. Given a path *p,* this is accomplished by considering all possible placements of *i* regenerators on *p.* Starting with *i=1,* the number of potential regenerators on p is increased by one until the first *i* is found that results in at least one placement of regeneration nodes with acceptable end-to-end signal quality across p. For this value of *i*, all possible placements of the *i* regenerators that rectify the signal on *p* are added to the list of candidate paths for serving the demand between the source and the destination of *p.*

Since the size of the ILP grows with the number of alternative paths considered, it follows that incorporating impairment-awareness into the problem increases the complexity of dimensioning. Depending on the particularities of the dimensioning problem to be solved, not all possible locations of the i regenerators that rectify the signal quality of a lightpath may be of interest. Particular to our problem definition, where we assume that each wavelength-selective switch is of dimension 2N – N being the number of fibers incident to the node – the consideration of any of the various possible placements of regenerators for a given path p is sufficient. As such, the incorporation of impairment-awareness does not increase the complexity of the ILP; only the pre-processing phase is slightly complex with worst case complexity $D \cdot k \cdot \sum_{i=0}^{d_G-1} \binom{d_G}{i}$, where d_G is the diameter of the network graph, *D* the number of demands and k the number of alternative shortest paths considered. For k being a small integer, the above complexity is O(D·2^{dG}), resulting theoretically to exponential preprocessing time. However, in most practical cases the length of a path will be shorter than the worst-case length (diameter). Also, for the network scales under study, the number of regenerators *i* required to sanitize an impaired lightpath will be a small fraction of the path length. Based on the above, the expected complexity of the preprocessing phase will hardly reach its theoretical upper bound and thus it is computationally feasible in most practical cases.

The result of the pre-processing phase described above is a set of at most *k* candidate paths for each demand *d*, together with a fixed position of *i* regenerators for each path p $(0 \le i \le n(p)-1, np : \textit{number of nodes of } p)$. The number of regenerators for each path *p* serving demand *d* is captured in the integer ILP constant r_{dp}. Let also *μ* stand for the equipment cost (transponder cost) of regenerating a single wavelength. The updated objective function of the IA-G-ND incorporating the additional cost of selective regeneration becomes:

$$Z = \sum_{e=1}^{|E|}(u_e \cdot \alpha_e) + \sum_{e=1}^{|E|}(y_e \cdot \beta_e) + \sum_{e=1}^{|E|}\sum_{c=1}^{W}(w_{ce} \cdot \gamma) + \sum_{n=1}^{|V|}\sum_{s=1}^{S}(t_{n,s} \cdot \varphi_s \cdot \theta_s \cdot W) + \sum_{d=1}^{D}\sum_{p=1}^{k}\sum_{c=1}^{W}(x_{dpc} \cdot r_{dp} \cdot \mu)$$

Apart from the additional pre-processing and the slight amendment applied to the objective function, the ILP formulation of the IA-G-ND problem is equivalent to the formulation of the G-ND problem (see 6Appendix B for a complete formulation of IA-G-ND).

The above formulation is complete in terms of consideration of costs under the already stated assumption that the dimensions of each switch are doubled to accommodate for connection termination/initiation or regeneration. In case the costs of adding switching redundancy is not acceptable by the network operator, the IA-G-ND should be adapted to consider all possible regenerator placements for each path. The reason for this is shown through the example of Figure 26, where two OXCs traversed by a lightpath p at wavelength λ_1 are illustrated. In the example, it is assumed that λ_1 is regenerated due to a lightpath other than p being impaired; thus, given that OXC_1 has four incident fibers, an 8x8 switch is required for switching wavelength λ_1 at OXC_1, while a 4x4 switch is sufficient for OXC_2, where no regeneration occurs. Assuming that the preprocessing phase indicates that both OXC_1 and OXC_2 are candidates for regenerating lightpath p, regenerating p at OXC_1 will not increase the dimensions of the λ_1 switch (2 spare ports), whereas the decision to regenerate p at OXC_2 will lead to additional cost due to installing an 8x8 instead of a 4x4 switch for λ_1 at OXC_2. **Consequently, if allocation of termination/regeneration ports is performed dynamically, placement of regenerators and network dimensioning should be carried out jointly to guarantee minimum cost.**

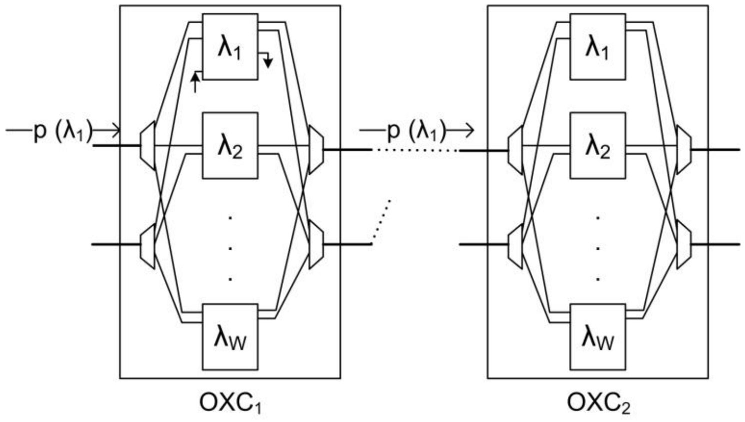


Figure 26: Two switches traversed by a lightpath p at wavelength λ_1, each having 4 incident fibers. λ_1 is already regenerated once at OXC_1 prior to installing p.

2.2.3.5 *On the Benefit of Impairment-aware Network Dimensioning*

As demonstrated in subsection 2.2.3.2, while selective regeneration based on the optical reach criterion works well in homogeneous networks and without considering impairments introduced at the switching nodes (e.g crosstalk, filter concatenation), it poses a serious trade-off between cost and effectiveness of regeneration in heterogeneous networks. In this section we aim at quantifying the benefit of impairment-aware network dimensioning compared to the distance-based approach. More precisely, we solve the IA-G-ND problem motivated by the Phosphorus EU topology and compare its incurred cost against the cost brought by the distance-based approach.

For a direct comparison to be meaningful the two approaches should produce results of equal effectiveness, i.e. produce a dimensioned Phosphorus EU optical network that satisfies all demands of the input traffic matrix. Specific to the distance-based approach, this translates to not allowing the approach to utilize lightpaths, whose BER is above the input BER threshold. The latter requirement, in conjunction with the fact that the distance-based approach does not per definition possess any information on the signal quality of a lightpath, lead us to use the distance covered by the shortest impaired path as the optical reach parameter in the distance-based approach. Essentially, this value constitutes a best-case scenario for the distance-based approach, since this is the value that yields the least number of regenerators without resulting in any impaired paths. For each input network configuration, we calculated the optimal optical reach value for the distance-based approach and used this value to parametrize this network-dimensioning approach.

To test the efficacy of IA-G-ND in a variety of heterogeneity levels, we randomly varied the noise figures (NF) of inline and node amplifiers deployed throughout the network. Starting with the homogeneous case, we used a uniform value of NF=6.5dB (level-1) for all amplifiers. Subsequently, we increased the degree of heterogeneity by modelling the noise figure of each amplifier as a normally distributed random variable with mean μ=6.5dB and standard deviation s=0 (level-1, homogeneous), s=0.5 (level-2), s=1 (level-3) and s=1.5 (level-4) respectively. Additionally and towards obtaining realistic results, we discarded random noise figure values that lied out of the [5,8]dB interval. Using the experimental setting outlined in subsection 2.2.3.2, we dimensioned the input network using IA-G-ND and the distance-based approach. For each heterogeneity level we use the number of regenerators installed by IA-G-ND as a criterion for statistical convergence: a sufficient number of repetitions are executed for the standard deviation of installed regenerators to become less than 15% of the mean.

The measured overhead of the distance-based approach in terms of mean number of installed regenerators over the IA-G-ND approach is shown in Figure 27. We first observe that even in the homogeneous case (level-1), the inherent inability of the distance-based approach to regenerate sub-paths that are shorter than the optical reach lead to substantial regeneration overhead compared to impairment-aware dimensioning. Throughout all heterogeneity levels tested, IA-G-ND resulted in at least 51% less regeneration costs with guaranteed (100%) signal quality for all installed lightpaths. Beyond the average statistical index, the same cost-efficiency trend is also evident for the 75% and 90% of the most cost-savvy regeneration effort among all samples (Figure 28). The ability of the impairment-aware approach to place regenerators with higher precision can be observed at the higher variability it exhibited compared to the distance-based approach: for the various random sample sets of amplifier noise figures, IA-G-ND adapts to the variability among the sets of a single heterogeneity level, while the distance-based approach does not have the means of capturing slight changes to signal quality that do not greatly affect the global optical reach value. This is clearly manifested in Figure 28 at the heterogeneity level-4, where the .75- and .9-quantile are approximately equal.

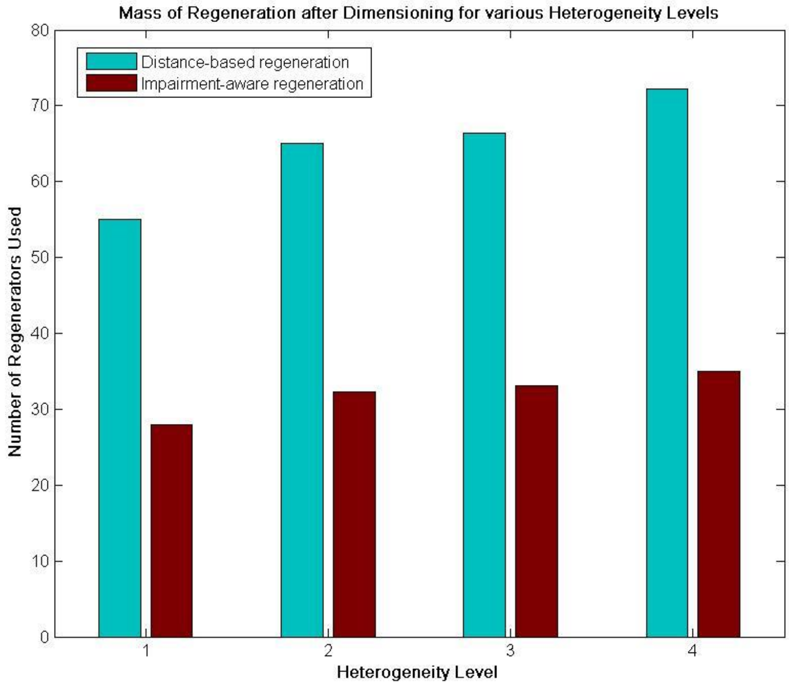


Figure 27: Benefit of impairment-aware network design in terms of mean number of regenerators required

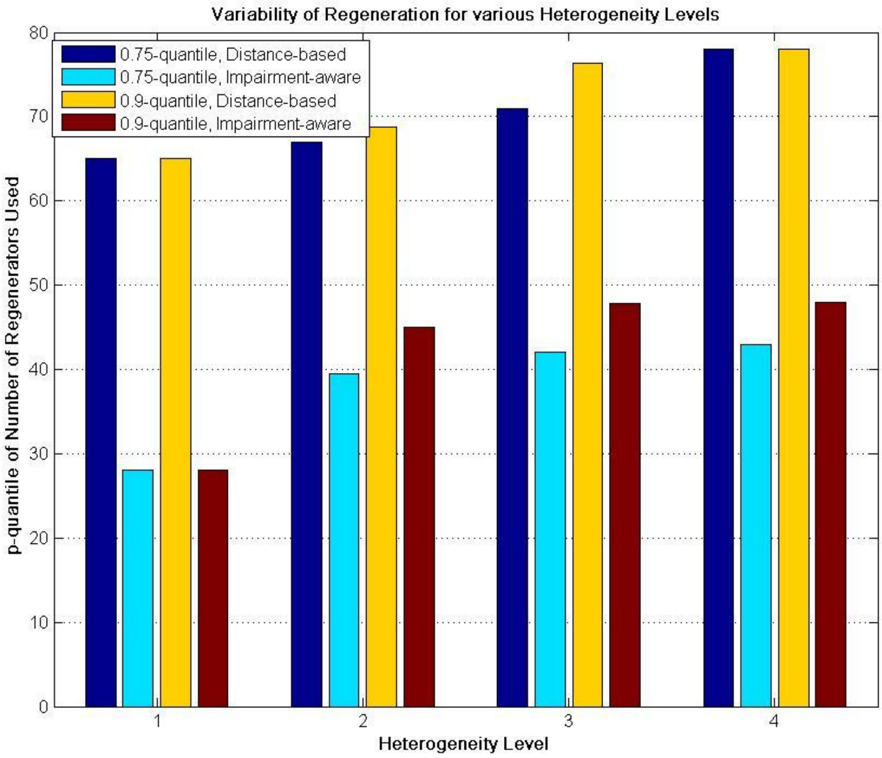


Figure 28: Variability of regeneration count samples for the two dimensioning approaches tested

2.2.3.6 *Discussion*

In this section we demonstrated the clear trade-off between regeneration cost and regeneration redundancy, when distance-based regeneration is employed in the network dimensioning phase. As a remedy, we presented an optimal method for joint network dimensioning and selective regeneration, which works on the basis of checking the signal quality across every lightpath to decide the placement of regenerators. Our novel approach achieved a significant reduction of the total regeneration cost of the dimensioned of EU Phosphorus testbed, independent of the heterogeneity of the network in terms of optical equipment. Although this study has only considered OEO regeneration, the ILP formulation we provided is also valid for the case of optical regeneration. In fact, we expect the IA-G-ND approach to result in large cost-savings compared to distance-based regeneration in the transparent case as well.

Part II

Research activity on optical Grids has traditionally focused on applications that require long-lived wavelength paths. The emergence of Grid computing has broadened the range of applications that are either enabled or otherwise greatly benefit from their implementation as Grid applications. For part of these applications, providing network services with wavelength granularity is not an efficient and scalable solution due to diverse traffic profiles and connectivity requirements.

Optical Burst Switching (OBS) is a suitable candidate for implementing a scalable network infrastructure to address the needs of emerging Grid networking services and distributed applications. Unlike optical wavelength switched networks, burst switching allows sharing of network resources with finer granularity across any timescale, allowing thus an adaptive network infrastructure able to support all application types, each with their own access, network and resource usage patterns.

Section 3 contains an introduction into the two optical switching primitives (wavelength-switching and burst switching) and a short primer on hybrid networks in terms of switching. Section 4 starts with identifying Grid application requirements that influence the selection of the switching technique that should be used to guarantee QoS at lower network cost. For each of the two switching alternatives, application characteristics are identified that speak for the selection of one or the other switching alternative. This qualitative analysis is backed by specific case studies, where the appropriateness of a switching technique for exemplary applications is shown. The part concludes with a quantitative evaluation of the two switching primitives through simulation.

3 Optical Switching Techniques

3.1 Circuit Switching

Traditionally, switching in optical networks has closely followed the connection-oriented paradigm, where for a specific session one (or more) fixed switched path(s) between source and destination are used to route data at the optical-layer throughout the entire session lifetime. The specific feature of circuit switching that differentiates it from other switching techniques is the granularity at which (re-)configuration of switches occurs; more precisely, in circuit switching, as the term implies, the finest switching granularity is the session, i.e. as soon as the configuration of switches is decided for a specific session, this configuration is used to route the entire amount of data exchanged throughout the entire session.

In summary in a circuit switched environment, each session is allocated a fixed fraction of the capacity on each link along its path. Consequently, circuit switching can offer fixed delays and guaranteed data delivery, while it suffers from poor resource utilization (bandwidth remains reserved even when there is no traffic between the endpoints), and its inefficiency for bursty traffic.

There are various reasons that justify the adoption of circuit-switching as the standard switching technique in optical networks. Historically, the configuration of switches to serve a connection request has been conducted manually and thus changing this configuration in smaller timescales as required by packet- or burst-switching was not an option. Even with the advent of automatically switched optical networks (ASONs), the overhead of switching at sub-session timescales is significant due to high delays in the reconfiguration of switches. Only recent advances in hardware technology have allowed the reconsideration of optical switching at sub-session timescales, such as burst switching (see next section).

The automatic setup of circuits is handled by special purpose protocols and machinery; the prominent architectures for this purpose are IETF's GMPLS [Mannie03] and ITU-T's ASON [G.8080]. The two complementary approaches are continuously standardized and are gradually reaching technological maturity. Since a prominent goal of the Phosphorus project is the extension of these standards with Grid-specific features, their workings have been already extensively described in past deliverables [D.2.1][D.2.2].

3.2 Optical Burst Switching

Optical burst switching (OBS) [Qiao99], [Turner99], [Varvarigos97] is a promising technology that has received an increasing amount of attention from both academia and industry. OBS can satisfy the rising bandwidth demands and reduces costs by overcoming the disadvantages of optical circuit switching (OCS) and optical packet switching (OPS). Although OCS is relatively easy to implement, it can not efficiently accommodate highly dynamic traffic. On the other hand, OPS requires optical technologies that are still immature, such as optical buffering and optical logic. OBS networks can be bufferless and can support users with different traffic profiles by reserving the bandwidth only for the duration of a burst. As Grid applications evolve, the need for user controlled network infrastructure is apparent in order to support emerging dynamic and interactive services. In an attempt to address this problem, a standardization document for Grids over OBS (GOBS) was submitted to the Open Grid Forum [GOBS].

The two main ideas in Optical Burst Switching are the assembly of variable sized data packets into bursts that are switched using a single label, and the decoupling of the transmission of the control header from the transmission of the data payload (out-of-band signalling). During the burstification process, multiple packets are aggregated into containers (referred to as data bursts) at the network ingress. Typically, an edge router maintains a separate (virtual) queue for each Forwarding Equivalent Class (or FEC), defined by the destination and QoS parameters. When a burstification threshold is reached and the burst has been formed, a control header, also called a burst header packet (BHP), is sent to the core network. The BHP is transmitted before the data payload in order to reserve the required bandwidth and configure the switches along the path. To do so, various signalling protocols have been proposed in the literature and will be reported later in this section. The separation between control and data maintains data transparency and leads to a better synergy of mature electronic technologies (which process the BHP) and advanced optical technologies (which handle the data burst). Figure 29 highlights the architecture of a typical OBS network, consisting of a cloud of optical core routers, organized as a flat mesh, with edge routers at the edges of the cloud. Core nodes are responsible for forwarding the burst to the proper destination edge node. Edge nodes are of two types: ingress (source) nodes and egress (destination) nodes. A node can be both a source and a destination. Burstification is performed at ingresses, where a burstification control unit (BCU) resides and coordinates the transmission of data and control packets. The opposite operation is performed at egress nodes, where bursts are converted back to packets.

In the following sections we describe the main aspects of the OBS networking paradigm in more detail.

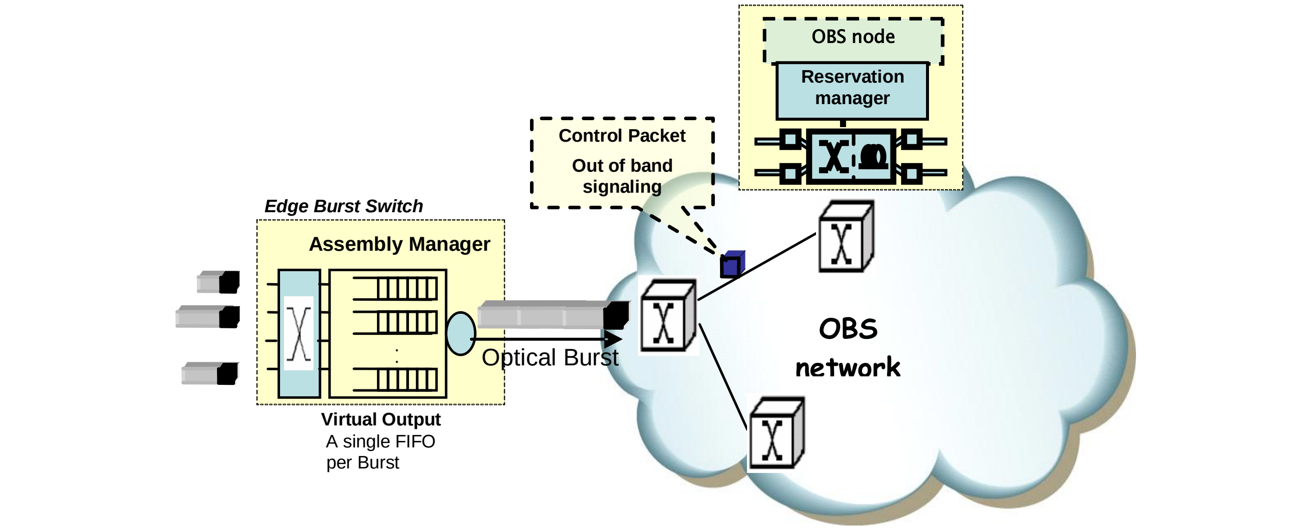


Figure 29: The architecture of a typical OBS network.

3.2.1 Burst Aggregation

OBS ingress nodes collect upper layer traffic and aggregate it into variable-size bursts based on destination addresses. When a burst arrives at the egress edge node it is disaggregated into original packets. Figure 30 illustrates the burst aggregation and desegregation processes.

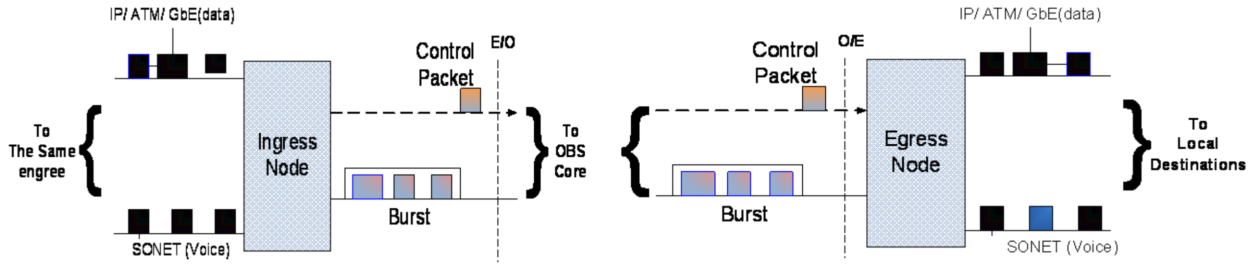


Figure 30: (a) Burst aggregation and (b) burst disaggregation processes

Burst aggregation algorithms could be timer-based or burst-length-based. In the timer-based algorithm (abbreviated T_{MAX} algorithm) [Ge00], all the packets that arrived in a pre set period of time (threshold T_{MAX}) are assembled into a burst. The time out value T_{MAX} for timer-based algorithm should be set carefully to ensure that packets waiting for the burst aggregation do not suffer intolerable levels of delay and at the same time not too many small bursts are generated resulting in a higher control overhead. Even though the time-based algorithm succeeds in limiting the average burstification delay, by limiting the maximum delay a packet can remain in the queue, it may generate very small bursts.

In the burst-length-based algorithm (abbreviated L_{MAX} algorithm) [Vokkarane03b], there is a threshold L_{MAX} on the minimum burst length. Once the threshold L_{MAX} (in fixed size packets or bytes, etc) is reached, the burst is created, its BHP is sent into the optical network and the data burst is transmitted after an appropriate offset time. This threshold should be optimized to ensure efficient resource utilization as long bursts may hold resource for long time and may cause high bursts loss, while short bursts may give rise to too many control packets. Bit padding can be used if there is not enough data to assemble the minimum size burst.

From above it is clear that while timer-based algorithms might create undesirable burst lengths, burst-length-based algorithms do not provide any guarantee on the aggregation delay that packets will experience. To address these deficiencies, a mixed timer/burstlength-based aggregation algorithm [Kantarci05] was proposed, where a burst can be sent out when either the burst length exceeds the desirable threshold or the timer expires.

Adaptive aggregation algorithms [Cao02] were also proposed where the time threshold or the burst length threshold or both can be adjusted dynamically according to real time traffic measurements. These algorithms improve the performance especially under strongly correlated input traffic but result in high operational complexity.

A possible benefit of burst aggregation is the differentiation of traffic classes by varying the burst aggregation time and the burst size [Ge00]. The burst aggregation algorithm also shapes the traffic by reducing the degree of self-similarity compared to the flow of the original higher-layer packets. As known self-similarity results in longer queuing delays than random (Poisson) traffic and higher packet losses, and therefore degrades network performance.

An average delay-based algorithm (abbreviated T_{AVE} algorithm), introduced in [Kaheel02], aims at controlling the average burstification delay by letting out the bursts, the moment the average delay of the packets that comprise it reaches a threshold T_{AVE}. This method guarantees a desired average burstification delay, and also tends to minimize packet delay jitter, which is particularly important for TCP performance.

3.2.2 Burst Reservation Protocols

Prior to the burst transmission a burst header packet (BHP) is created and sent to the destination. The BHP is typically sent out of band over a separate signalling wavelength and processed electronically at intermediate OBS routers. The BHP contains information about the corresponding burst is transmitted to reserve the required transmission and switching resource. Data bursts are typically not buffered at intermediate OBS routers and remain in the optical domain end-to-end and as they transit the network core.

After generating a burst using the algorithms discussed above, the burst is delayed for an offset time before being transmitted to allow its corresponding BHP to set necessary reservations at the intermediate nodes. Burst reservation could be implemented using one way (tell-and-wait) or two way (tell-and-go) signalling protocols.

In one way signalling, the burst is transmitted after an offset time without waiting for positive acknowledgement assuming that entire path has been successfully established and necessary resource reservation are made to allow the burst to be switched in a cut through manner. The offset time should be large enough to ensure that the burst may not arrive at a node before completely setting up switching configuration. One way OBS gives much shorter setup time and better throughput performance but it may result in burst loss if the reservation at an intermediate node is not successful (assuming there are no Fiber Delay Lines (FDLs) to store the burst).

In two ways signalling, the burst would not be transmitted unless an end-to-end connection is guaranteed between ingress and egress nodes of the network. There are two types of two-way protocols, centralized and distributed. In the centralized architecture, setup requests are sent to a centralized scheduler, where they are queued based on their destination address and assigned available wavelength. A positive ACK is sent to OBS user to transmit its burst. This approach guarantees burst delivery and requires a simple node design. However, network wide information is required and the overall reconfiguration speed and possible scalability are limited. In the distributed architecture, a source that has a burst to transmit first tries to reserve the appropriate resources from source to destination by sending a request message. Intermediate nodes receiving this message reserve the bandwidth on the desired outgoing links starting at the time the burst is expected to arrive at these nodes, and for duration equal to the duration of the burst. If this process is successful on all the links along the path, an ACK is sent back to the source, which then sends out the burst; otherwise, a NAK is returned to release the previously reserved bandwidth, and initiate the retransmission of the request message.

An example of a tell-and-wait protocol is the Efficient Reservation Virtual Circuit (ERVC) protocol proposed in [Varvarigos98], while recent research efforts include WR-OBS [Dueser02] and the Efficient Burst Reservation Protocol (EBRP) [Christodou06]. The Ready-to-Go Virtual Circuit (RGVC) protocol [Varvarigos97] is an early attempt for a one-way reservation scheme in optical networks. In the recent years, one-way reservation schemes have received increasing attention. The Horizon protocol has been presented in [Turner99] and Just Enough Time (JET) protocol has been proposed in [Qiao99]. The Just-In-Time (JIT) protocol proposed in [Wei00] is a centralized protocol as it requires each burst transmission request to be sent to a central scheduler, which then informs the requesting node of the appropriate time to transmit the burst. Since centralized protocols are neither scalable nor robust a distributed version of JIT was presented in [Baldine02].

Upon reception of a BHP, the OBS core nodes schedule their resources according to the information in BHP. Resources can be allocated by using explicit setup or estimated setup. In explicit setup, a wavelength is reserved, and the node switch is configured immediately after the processing of the BHP. In estimated setup (which is also referred to as advance reservations), the OBS node delays the reservation and configuration until the actual burst arrives. The offset time is included in control packet and updated at each node. In this way the node will know the burst arrival time. The allocated resources can be released after the burst has passed through by using either explicit release or estimated release. In explicit release, the end of burst transmission is indicated by sending trailing control packet. In estimated release, an OBS node knows exactly the end of the burst transmission from the burst length, and therefore can calculate when to release the occupied resources. Based on these classifications four possibilities could be obtained: (i) explicit setup/explicit release, (ii) explicit setup/estimated release, (iii) estimated setup/explicit release, and (iv) estimated setup/estimated release. Estimated schemes result in shorter reservation periods and thus higher network throughput compared to explicit schemes. But they have complicated implementations and their performance greatly depends on offset estimation. On the other hand, explicit schemes result in longer reservation times and therefore higher burst loss probability, but they are less complicated to implement compared to estimated schemes.

Finally, when an intermediate node receives the BHP setup packet there is the issue of which wavelength (channel) to choose on the decided outgoing link to serve the burst. This is usually referred to as the link scheduling problem. Link scheduling algorithms can be classified into two categories: with [Xu03] and without [Turner99] void filling (VF). In the link scheduling algorithm of [Turner99], a scheduler assigns a burst to the channel with the latest horizon (latest scheduled burst on the channel) as long as it is earlier than the arrival time of the burst. The main advantages of this algorithm are its minimal memory requirements and the speed of execution. However, the Horizon scheduling algorithm does not try to exploit the void intervals, resulting in low bandwidth utilization and a relatively high loss ratio. Various algorithms with void filing are presented in [Xu03]. One such algorithm is the LAUC-VF (latest available unscheduled channel with void filling) algorithm, which yields better bandwidth utilization and loss rate than the Horizon algorithm, but requires more time to execute.

3.2.2.1 *Link Utilization Profiles*

In the optical burst routing paradigm, each node needs to keep a record of the capacity reserved on its outgoing links and wavelengths as a function of time [Varvarigos98], [Xu03] in order to perform channel scheduling and reservation.

Assuming that each session or burst reserves bandwidth equal to the wavelength capacity (C_w) for a given time duration, the *utilization profile* $U_w(t)$ of a wavelength *w* is a stepwise binary function with discontinuities at the points at which reservations begin or end, and is updated dynamically with the admission of each new session or burst. We define the *capacity availability profile* of wavelength *w* of capacity C_w as $C_w(t)=C_w-U_w(t)$ (Figure 31)

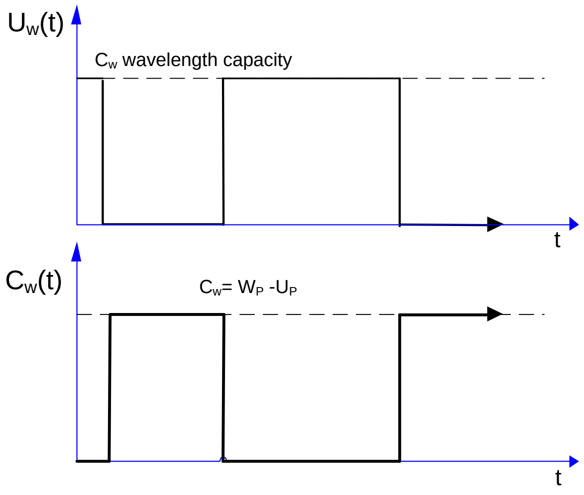


Figure 31: Step-wise utilization profile of a wavelength.

3.2.3 Routing and scheduling algorithms for OBS networks

In the previous section efficient protocols for capacity reservations were presented. A major issue in OBS networks is how to select the paths to be followed and the times the bursts should start transmission, so as to avoid the burst contentions in the core and balance the traffic load of the network.

Contention avoidance and load balancing for OBS networks can be implemented using open-loop or close loop (also called feedback-based) algorithms. Open-loop algorithms use traffic shaping at the ingress node [OFC02], [Farahmand04] to reduce incoming traffic burstiness. Since the ingress node does not receive any feedback on the network state, open-loop algorithms cannot respond to traffic changes in the network. In contrast, close loop algorithms continually monitor global network load and resource availability information, and try to avoid contention by varying accordingly the data flows [Farahmand04], [Wen05], [Thodime03], [Zapata03], [Yang05b], [Teng05], [Varvarigos08]. A limitation of these algorithms is that when the bandwidth-delay product of the network is large, the edge nodes' responses tend to be slow, potentially leading to traffic oscillations and unstable network behaviour.

In [Farahmand04b] the authors describe a feedback-based contention avoidance scheme for OBS networks, where core switches send explicit messages to edge nodes requesting them to reduce their transmission rate on congested links. The integration of contention resolution in the GMPLS framework is investigated in [Wen05], where a load balancing technique is proposed, and is divided in two parts: (i) traffic engineering is used at the edge of the network to reduce contention and (ii) wavelength conversion and fibre delay lines are used for buffering in the core. In [Thodime03] fixed alternate paths and a dynamic route calculation technique are proposed to reduce contention. In [Zapata03] the authors introduce a dynamic Wavelength Routed OBS (WR-OBS) architecture where centralized control is employed to provide resource reservation efficiency, low delay, and QoS differentiation. In this algorithm the decisions are taken assuming advance reservations are employed in the underlying OBS network. The authors in [Yang05b] present routing strategies that utilize different types of network state information to rank alternative paths. In addition, hybrid algorithms that combine these strategies with static or dynamically adjusted weights are proposed. Taking a different approach, [Teng05] presents a formulation of the path selection problem in OBS networks as an integer linear optimization problem by making some simplifying assumptions. This formulation tries to optimise the throughput of the whole network by scheduling the bursts accordingly, exploiting the concept of advance reservations to a great extend.

A recent approach in this field is a multi-cost routing and scheduling algorithm presented in [Varvarigos08]. The algorithm selects the paths to be followed by the bursts and the times when the bursts should start transmission from their source so as to arrive at their destination with minimum delay, addressing the burst routing and advance reservation planning problem jointly. The algorithm can function both with one- and two-way reservation schemes that have to employ advance reservations (algorithms that employ estimated setup) in order to enable the efficient routing and scheduling of the bursts.

3.2.4 Novel Assembly Techniques and Fast Reservation Protocols Based on Traffic Prediction

The end-to-end delay over an OBS network mainly consists of four components: (i) the burst assembly delay at the ingress node, (ii) the path setup delay caused by the BHP, (iii) the burst transmission time, and (iv) the propagation delay in the core network. The two last delay components depend on the path selected and the available bandwidth on the path and cannot be reduced through clever design of the assembly algorithms and the signalling protocols. Our work [Sideri07], [Seklou08], [Seklou08b] focuses on the first two delay components, and consists of two main contributions. First, it proposes new burst assembly algorithms that minimize the burst assembly delay, for a given average size of the bursts produced. Second, it uses pipelining techniques to reduce the combined duration of the burst assembly and path setup time and the overall end-to-end delay. The end-to-end delay is a crucial QoS parameter for a number of applications such as voice, videoconferencing and real time applications. The burst dropping probability is another important QoS parameter of interest in this work. Finally, the size of the bursts produced is also important in determining the control overhead posed on the network.

More specifically, in Section 3.2.4.1 we propose and evaluate several novel burst aggregation schemes that use traffic prediction in order to maximize the average length of the bursts produced for a given average burstification delay, or, alternatively, to minimize the average burstification delay for a given average burst length. Prediction of traffic characteristics has previously been examined in [Xu03], [Ostring01], [Morato01]. In [Xu03] it is demonstrated that despite the long-range dependence of Internet traffic, which would lead us to expect that we must look deep into the "past" for a precise estimation, a prediction filter of small order is sufficient for good performance. We use traffic prediction in order to estimate the number of packets that will arrive at the assembly queue in the near future and determine if it would be beneficial for the burst assembly process to wait for these packets, or the burst should be sent immediately. The performance measure we use to compare the algorithms proposed is the Average Packet Burstification Delay to Burst Size ratio (DBR) defined as:

$$DBR = \frac{\text{Average Packet Burstification Delay}}{\text{Average Burst Size}}$$

We find that two of the proposed schemes improve burstification efficiency over the previously proposed schemes, by reducing the average burstification delay by up to 33% for a given size of the bursts produced, compared to previous schemes.

Following the introduction of the new burst assembly schemes, we turn our attention to signalling protocols for connection establishment and resource reservation in OBS networks to obtain fast reservation protocols, so as to reduce the second delay component of OBS networks that is the path setup delay. In Section 3.2.4.2 we propose fast reservation (FR) schemes that can be combined with the burst assembly algorithms introduced earlier in this work, to further reduce the burst pre-transmission delay (the “time offset”). This is achieved by using one or two prediction filters to estimate the length of the burst and/or the time needed for the burstification process to complete, and by pipelining the reservation and the burst assembly processes. We use the TAG scheme, since it incurs a smaller pre-transmission delay at the ingress node, but in contrast to standard OBS signalling protocols, in our work the BHP is sent to the core network before the burst assembly is completed. Intermediate nodes use the estimated values for the burst length and assembly completion time, instead of the actual values that are not yet known at the time BHP is sent, in order to reserve bandwidth for the intervals the burst is expected to pass from these nodes. Estimating the length of the burst is required in order to reserve the required resources in the core network for the right duration for the burst's all-optical transmission. Estimating the duration of the burst assembly process is required in order to determine the time these reservations should start at the core nodes. Our goal is (i) to reduce the end-to-end delay of a data burst, by minimizing the burst pretransmission time, while (ii) using bandwidth efficiently by reserving resources for a duration that is close to the minimum possible. In [Morato01] the authors evaluate the use of a linear prediction filter along with the T_{MAX} burst assembly algorithm to reduce the burst pre-transmission delay in a way similar to the present work. Our results extend the work in [Morato01] to show that prediction can also be used along with the L_{MAX}, T_{AVE} and the other burst assembly algorithms, reducing the end-to-end delay of the bursts in these schemes as well.

3.2.4.1 *Novel Burst Assembly Algorithms*

In this section we focus on the burst assembly process, and propose burst aggregation algorithms that try to minimize the average burstification delay, for a given average size of the bursts produced. Generally, the burst assembly process at an edge node starts with the arrival of the first packet at an empty queue and continues until a predefined threshold is reached. Different assembly strategies define differently this threshold, and try to balance between two objectives: the burstification delay and the size of the bursts produced. Short burstification delays and large burst sizes are desirable, in order to reduce, respectively, the total end-to-end delay and the number of bursts along with the processing overhead they pose on the core nodes. These objectives, however, contradict each other (Figure 32), since increasing the burst size also increases the burstification delay. A burst assembly algorithm, therefore, should be judged based on how well it performs with respect to one of these two performance metrics of interest, for a given value of the other performance metric. In Figure 32, the burst assembly algorithm used determines the curve that relates the average burst size to the average burstification delay. Given a burst assembly algorithm, choosing the desired balance between the burstification delay and the burst size (that is, the exact point on the curve of Figure 32 at which the system operates) depends on the QoS requirements of the users, and the processing and buffering capabilities of the backbone nodes.

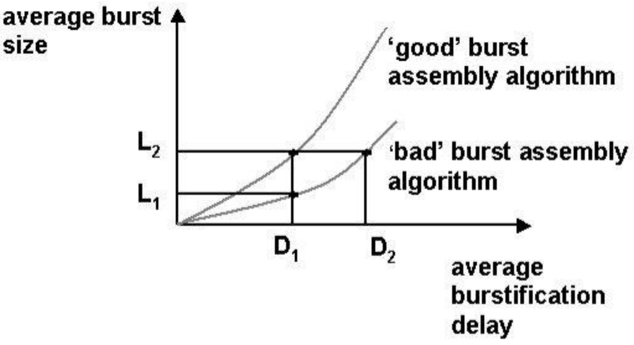


Figure 32: Performance of a "good" and a "bad" burst assembly algorithm. A "good" algorithm produces bursts of larger average size, for a given average burstification delay (alternatively, they result in smaller average burstification delay for a given average size of the bursts produced).

In our proposed schemes, we assume that the time axis is divided into time frames of equal duration τ (see Figure 33). During a frame, an edge OBS node assembles the packets arriving with the same destination address and the same QoS requirements (that is, the same FEC) into a burst. We denote by $N(n)$ the number of packet arrivals during the n^{th} frame. At the end of each frame, a decision is taken about whether the burst should be sent out immediately and the assembly of a new burst should start, or the edge node should wait for another frame in order to include more packets in the current burst. This decision is taken by using a linear prediction filter (described in Section 3.2.4.3) to estimate the expected number $\hat{N}(n+1)$ of packet arrivals in the following frame $n + 1$, and checking if a specific criterion (different for each algorithm proposed) is fulfilled. This criterion tries to quantify if the increase in the burst length expected by waiting for an extra frame is significant enough to warrant the extra delay that will be incurred.

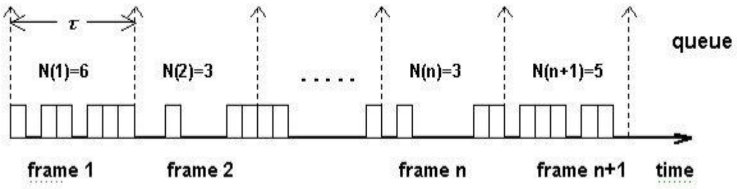


Figure 33: Time frame structure. At the end of each frame n the algorithm decides if it should send out the burst immediately, or it should wait for another frame. *N*(n) is the number of packet arrivals during the nth frame.

The following subsections describe the proposed burst assembly algorithms, while the corresponding performance results and comparison between the schemes are deferred until Section 3.2.4.4.

Fixed Additional Packets Threshold Algorithm (N_{MIN} algorithm)

In this scheme, we define a lower bound N_{MIN} on the number of future arrivals above which we decide to wait for an extra frame before assembling the burst. At the end of frame *n*, the estimate $\hat{N}(n+1)$ produced by the linear predictor is compared to the threshold N_{MIN}, and if it is smaller than that, the burst leaves the queue immediately; otherwise it waits for another frame to be completed, at the end of which the same procedure is repeated. Therefore, the burst is sent out at the end of the n^{th} frame if and only if:

$$\hat{N}(n+1) < \mathrm{N}_{\mathrm{MIN}} . \qquad (1)$$

Proportional Additional Packets Threshold Algorithm (aL algorithm)

In this proposed scheme, instead of using a fixed threshold value N_{MIN}, a fraction of the current length of the burst is used as the threshold. If *α* is the multiplicative parameter, the burst is completed at the end of frame *n*, if and only if

$$\hat{N}(n+1) < a \cdot L(n) , \qquad (2)$$

where *L*(*n*) is the burst length at the end of the n^{th} frame, and $\hat{N}(n+1)$ is the predictor's estimate for the number of packets expected during the following frame *n* + 1.

Average Delay Threshold Algorithm (T_A algorithm)

This method tries to improve on the average-delay-based algorithm proposed in [Christodou07], which computes a running average of the packet burstification delay and lets out the burst, the moment the average delay of the packets that comprise it reaches a threshold T_{AVE}. The algorithm in [Christodou07] has two drawbacks: a) computing the running average introduces considerable processing overhead, and b) bursts may not be sent out at the optimal time, since the running average is non-monotonic in time and could decrease in the future due to new packet arrivals. The T_A algorithm addresses these drawbacks using traffic prediction. At the end of each frame, it estimates the average burstification delay we expect to have at the end of the following frame, and launches the burst if this estimate exceeds some threshold value T_A.

The Average Packet Delay *D*(*n*) of the packets in the burst assembly queue at the end of frame *n* is defined as

$$D(n) = \frac{\sum_{i=1}^{L(n)} T_i(n)}{L(n)} ,$$

where *L*(*n*) is the burst size (in packets) at the end of frame *n*, $T_i(n) = n \cdot \tau - t_i$ is the delay of the i^{th} packet from the moment it enters the queue until the end of n^{th} frame, *τ* is the duration of the frame, and t_i is the arrival time of the i^{th} packet. Alternatively and more easily, we can compute *D*(*n*) using the recursion:

$$D(n)=\frac{L(n-1)\cdot D(n-1)+L(n-1)\cdot\tau+\sum_{i=1}^{N(n)}T_i(n)}{L(n-1)+N(n)}, \tag{3}$$

where $N(n)$ is the number of packet arrivals during frame n. If a burst was sent out at the end of the $(n - 1)^{th}$ frame, we take $L(n - 1) = 0$ in Eq. (3).

To obtain an estimate $\hat{D}(n+1)$ of the Average Packet Delay at the end of frame n + 1 we assume that the $\hat{N}(n+1)$ packets estimated by the predictor to arrive by the end of frame n + 1 will have an average delay of $\tau/2$. Using Eq. (3), the estimated Average Packet Delay $\hat{D}(n+1)$ at the end of frame n + 1 is

$$\hat{D}(n+1)=\frac{L(n)\cdot D(n)+L(n)\cdot\tau+\hat{N}(n+1)\cdot\frac{\tau}{2}}{L(n)+\hat{N}(n+1)}. \tag{4}$$

A burst is completed at the end of the n^{th} frame if and only if

$$\hat{D}(n+1)>T_A, \tag{5}$$

where T_A is the predefined threshold value.

Average Delay to Burst Size Ratio Improvement Algorithm (L_{MIN} algorithm)

The proposed Average Delay to Burst Size Ratio Improvement algorithm with threshold L_{MIN} (abbreviated L_{MIN} algorithm) uses traffic prediction to compute an estimate $\widehat{DBR}(n+1)$ of *DBR* (defined in the introduction of Section 3.2.4) at the end of frame n + 1, and decides that the burst is completed, if this estimate is worse (larger) than the current value *DBR(n)*. The average burstification delay to burst size ratio *DBR(n)* at the end of frame n is defined as:

$$DBR(n)=\frac{D(n)}{L(n)}=\frac{\sum_{i=1}^{L(n)}T_i(n)}{L^2(n)}.$$

Alternatively, and more easily, *DBR(n)* can be found recursively as:

$$DBR(n)=\frac{L(n-1)\cdot D(n-1)+L(n-1)\cdot\tau+\sum_{i=1}^{N(n)}T_i(n)}{(L(n-1)+N(n))^2}. \tag{6}$$

The Estimated Average Packet Burstification Delay to Burst Size ratio $\widehat{DBR}(n+1)$ at the end of frame n + 1 can be found as:

$$D\hat{B}R(n+1) = \frac{L(n)\cdot D(n) + L(n)\cdot\tau + \hat{N}(n+1)\cdot\frac{\tau}{2}}{(L(n)+\hat{N}(n+1))^2} \,. \tag{7}$$

The algorithm decides that a burst is completed and should be sent out at the end of frame *n* if and only if:

$$D\hat{B}R(n+1) < DBR(n) \,, \tag{8}$$

$$\text{and} \;\; L(n) > L_{MIN} \,. \tag{9}$$

During the first frames that follow a burst assembly completion, there is a great likelihood that the right term of Eq. (8) will be quite small, making it difficult to fulfil. The threshold L_{MIN} is used as a lower bound on the length of the bursts, and also makes the algorithm parametric (as with the previous algorithms examined) so that the desired trade-off between the average burst size and the average packet burstification delay can be obtained.

Simulation results on the performance of the preceding burst assembly algorithms are presented in Section 3.2.4.4. In what follows, we turn our attentions to signalling protocols for reducing the second delay component incurred in OBS networks, which is the path setup delay.

3.2.4.2 *Fast Reservation Protocols*

In this section, we look into to fast reservation (FR) protocols that can be used with the T_{MAX}, L_{MAX}, T_{AVE} algorithms, or any of the other novel burst assembly algorithms introduced in Section 3.2.4.1, to reduce the combined duration of the burst assembly and path setup time, further reducing the overall end-to-end delay. Such a scheme was first presented in [Morato01], where a fast reservation protocol that uses the T_{MAX} assembly scheme and a prediction filter was proposed. In this section we extend this scheme so that it can be combined with the other burst assembly algorithms.

The proposed FR protocols use one or two linear prediction filters to estimate the burst length and/or the time needed for the burstification process to complete. In contrast to standard OBS signalling protocols, in the FR schemes the BHP is sent to the core network *before* the burst assembly process is completed, in order to reserve the appropriate resources. To do so, it uses the estimated values of the burst length and assembly completion time, instead of the actual values that are not yet known at the time BHP is sent, to reserve bandwidth at each intermediate node for the interval the burst is expected to pass from that node.

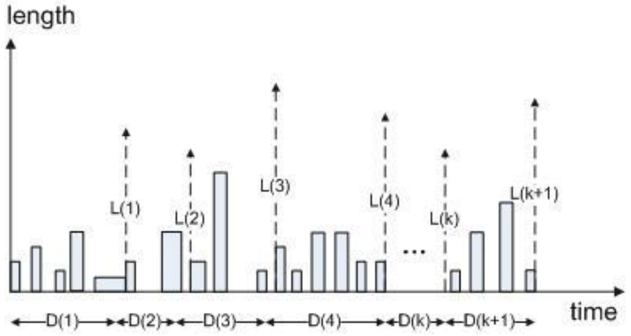


Figure 34: Prediction of the burst size and burst assembly duration is performed based on the k previous burst lengths and assembly durations. *L*(k) denotes the size of the kth burst (in bits) and *D*(k) is its assembly process duration.

We let *L*(*k*) be the size of the *k*th burst (in bits) and *D*(*k*) be its assembly process duration (Figure 34). If both were known at the beginning of a burst assembly period, we could start the reservation process at that time, reducing the overall delay. Since *L*(*k*) and *D*(*k*) are not known in advance, the idea is to start the reservation process before the burst is assembled, using estimates of these variables. In the time-based burst assembly algorithm *D*(*k*) is fixed and equal to T_{MAX} (therefore, we only have to estimate *L*(*k*)), while in the length-based algorithm *L*(*k*) is fixed and equal to L_{MAX} (therefore, we then only have to estimate *D*(*k*)). In the average delay-based algorithm, as well as in all the burst assembly algorithms introduced in Section 3.2.4.1, both the burst length *L*(*k*) and the burst assembly duration *D*(*k*) vary and have to be estimated.

Fast Reservation for the T_{AVE} scheme

The signalling used by the fast reservation (FR) protocol for the T_{AVE} assembly scheme is illustrated in Figure 35 (the cases of the burst assembly algorithms of Section 3.2.4.1 are similar, while those of the T_{MAX} algorithm and the L_{MAX} algorithm are simpler): upon the beginning of a new burst assembly period we use two Least Mean Squares (LMS) filters to predict burst related values. Using these predictions, a BHP is sent at the beginning of the burst assembly process to reserve in advance the required resources, instead of waiting for the burst assembly to complete.

Specifically, the first LMS filter is used to obtain a prediction of the length $\hat{L}(k)$ (in bits) of the *k*th burst to be formed; this value is included in the BHP and is used to reserve bandwidth for a duration that is close (if the prediction is accurate) to the burst's real transmission duration. The second filter produces a prediction of the assembly process duration $\hat{D}(k)$, which is also included in the BHP, and is used to reserve bandwidth at each intermediate link starting at the correct (if the prediction is accurate) time instant. To reduce the effects of prediction errors in the burst length, we add a small margin *δ* in the estimated burst length $\hat{L}(k)$. This is done in order to reduce the probability that bandwidth is reserved for less time than the actual burst duration: no matter how accurate is the filter, the actual length *L(k)* of the *k*th burst will be larger than $\hat{L}(k)$ approximately half of the time, which would be unacceptable, however, *L*(*k*) will be smaller than $\hat{L}(k)+\delta$ with high probability if the prediction filter is good and *δ* is large enough. Similarly, to reduce the effects of prediction errors in the assembly process duration, we subtract a small margin *ε* from the estimated duration $\hat{D}(k)$ of the burst assembly process. This is because $\hat{D}(k)$ is used to calculate the time at which reservations at intermediate links should start, and in case of uncertainty, it is safer to start reservations a little earlier than the predicted starting time. By using these safety margins, the reservation starts earlier than the expected time by *ε* and finishes later than the expected time by *ε* + *δ/C*, where *C* is the reserved bandwidth. This way we can be reasonably certain that the burst will find capacity already reserved for it when it arrives at a node. Therefore, bandwidth is reserved at each intermediate node for the time period:

$$\left[\hat{D}(k)-\varepsilon, \hat{D}(k)+\varepsilon+\frac{\hat{L}(k)+\delta}{C}\right], \tag{10}$$

where times are relative to the arrival time of the BHP at each node. Note that if the estimators of *L*(*k*) and *D*(*k*) are unbiased, capacity is reserved for a burst for time 2*ε*+*δ/C* more than the minimum required, on the average. The inefficiency caused by this is negligible if *ε* and *δ* are small.

When burst assembly is completed, the predicted values $\hat{D}(k)$ and $\hat{L}(k)$ are compared with the real values of *D*(*k*) and *L*(*k*). The ingress node sends the burst after a small (pre-transmission) interval t_x, calculated so as to compensate for predictions errors, as will be described shortly. If the burst is sent after time t_x, the time period the burst actually traverses the network is:

$$\left[D(k)+t_x, D(k)+t_x+\frac{L(k)}{C}\right], \tag{11}$$

where *L*(*k*) is the burst's actual length and *D*(*k*) its assembly duration. In order for the in advance reservation to be successful, the reservation period must contain the burst's actual transmission period. That is, the reservation at any core node should start before the burst arrives and should finish after the burst's departure. So, based on Eq. (10) and Eq. (11) the following conditions must hold:

$$t_x + D(k) > \hat{D}(k) - \varepsilon \;, \tag{12a}$$

and $$t_x + D(k) + \frac{L(k)}{C} < \hat{D}(k) + \varepsilon + \frac{\hat{L}(k)+\delta}{C} \;. \tag{12b}$$

The pretransmission time t_x can be chosen equal to

$$t_x = \max(\hat{D}(k) - \varepsilon - D(k), 0) \;,$$

so as to minimize pre-transmission delay, while always satisfying Eq. (12a). In that case,

$$\Pr(t_x = 0) = \Pr(D(k) > \hat{D}(k) - \varepsilon) \,,$$

and the pre-transmission delay will be zero with high probability.

A sufficient set of conditions to satisfy Eq. (12b) is

$$L(k) < \hat{L}(k) + \delta \, , \qquad (13a)$$

and $$D(k) < \hat{D}(k) + \varepsilon \, , \qquad (13b)$$

which will both be valid with high probability, provided that *ε* and *δ* are sufficiently large. In that case the reservation will be successful, in the sense that bandwidth will be reserved for a duration that is close to (and larger than) the burst's real transmission duration. Also, the pre-transmission delay will be zero with high probability.

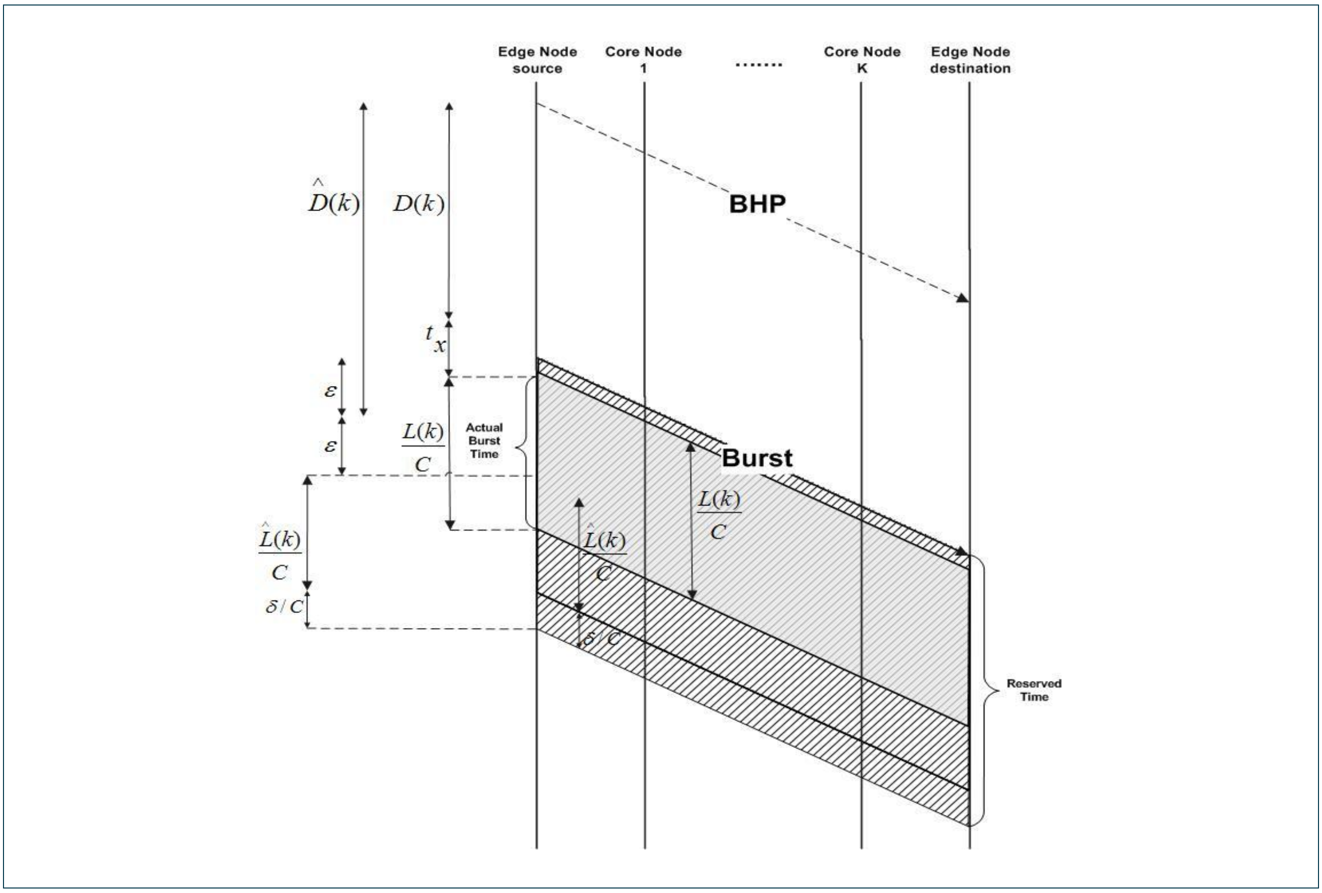


Figure 35: A successful reservation for the T_{AVE} burst assembly algorithm, using two predictive filters. One filter predicts the burst length $\hat{L}(k)$ and the other the burst assembly duration $\hat{D}(k)$. The figure illustrates the time instants at which reservations start and finish at each node, and the time intervals the burst actually passes from a node.

If Eq. (12a) and (12b) cannot be simultaneously satisfied for any choice of t_x, the transmitted BHP is a failure (Figure 36). We then have to transmit a new BHP to cancel the old reservation and perform a new one with the actual burst size *L*(*k*) and the right reservation starting time. It is evident that the failed reservation does not result in a burst loss, but only some loss of efficiency in the small interval between the reservation and the cancellation, where the resource remains idle.

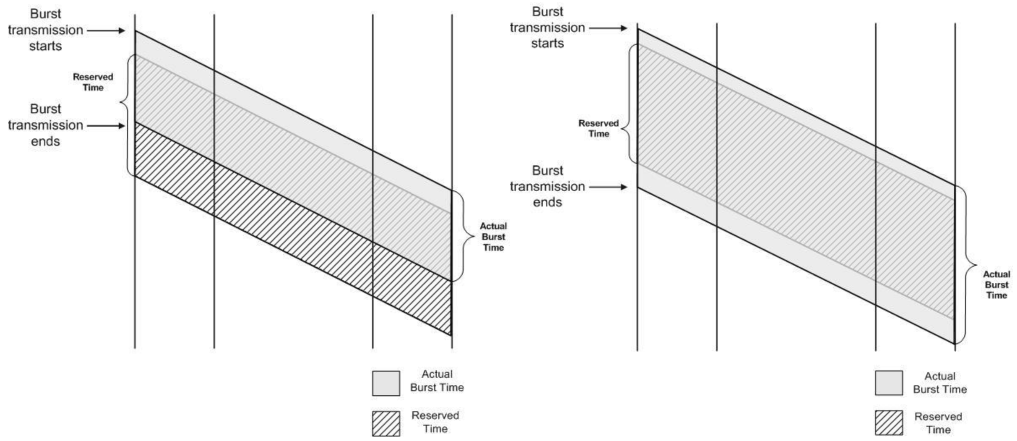


Figure 36: Failed reservations for the $\mathrm{T}_{\mathrm{AVE}}$ assembly algorithm. (a) Illustrates the case burst transmission starts earlier than the reservation, (b) illustrates the case the burst length exceeds the reserved duration.

Fast Reservation for the $\mathrm{L}_{\mathrm{MAX}}$ scheme

In the case of the $\mathrm{L}_{\mathrm{MAX}}$ assembly algorithm, the burst length *L*(*k*) is fixed and known a priori. In that case reservations are performed as in Fig. 5, but with $L(k) = \mathrm{L_{MAX}}$ and $\delta = 0$. A prediction filter is used to obtain the estimate $\hat{D}(k)$ of the k^{th} burst assembly duration, on which we use a small safety margin ε to compensate for the case the prediction turns out to be larger or smaller than the actual value. The BHP is sent to reserve the necessary resources starting a little earlier than the predicted time, without waiting for the burst assembly to complete. Specifically, the BHP, upon its arrival at a core node, reserves bandwidth *C* for the time period:

$$\left[\hat{D}(k)-\varepsilon, \hat{D}(k)+\mathrm{L_{MAX}}/\mathrm{C}+\varepsilon\right], \qquad (14)$$

relative to its arrival time at that node. When the burst assembly is completed, the actual assembly duration *D*(*k*) is compared to $\hat{D}(k)-\varepsilon$.This comparison is performed in order to ensure that the reservation of the resources in the network starts at the right time. The pretransmission time is again chosen according to

$$t_x = \max(\hat{D}(k)-\varepsilon-D(k),0).$$

Provided that

$$D(k) < \hat{D}(k) + \varepsilon \,, \tag{15}$$

the reservation made by the BHP is successful (if $\hat{D}(k) - \varepsilon < D(k) < \hat{D}(k) + \varepsilon$, we additionally have $t_x = 0$). Otherwise, the pretransmitted BHP is a failure and we have to transmit a new BHP to cancel the old reservation and perform a new one.

Fast Reservation for the T$_{MAX}$ scheme

In the case of the T$_{MAX}$ assembly algorithm, the burst assembly duration is known a priori, since *D*(*k*)= T$_{MAX}$. In that case reservations are performed as in Fig. 5 but with $\hat{D}(k) = \mathrm{T_{MAX}}$ and $\varepsilon = 0$. A filter is used to predict the *k*th burst length $\hat{L}(k)$, and bandwidth is reserved for time $(\hat{L}(k) + \delta)/C$. When the time threshold T$_{MAX}$ is reached, the burst assembly is completed, and the actual burst length *L(k)* is compared with the predicted length. If

$$\hat{L}(k) + \delta > L(k) \,, \tag{16}$$

the pretransmitted BHP reserves capacity for sufficient duration and the reservation is successful. Otherwise, the BHP is considered to be a failure and a new BHP must be sent to cancel the old reservation and perform a new one for the actual burst size *L*(*k*).

Fast Reservation and Minimum Separation

The transmission of the BHP has to precede the transmission of the burst by at least a time offset equal to t_0, where t_0 is a parameter chosen to account for the extra processing delays the BHP (which is processed electronically) encounters at intermediate nodes when compared with the processing delays encountered by the burst (which is switched all-optically). For example, if t_{el} is the time it takes for a core node to process electronically the BHP and t_{ao} is the time it takes for the core node to switch (all-optically) a burst from an input to an output port, we can choose

$$t_0 = h \cdot (t_{el} - t_{ao}) \,, \tag{17}$$

where *h* is the number of hops on the path.

If the estimate $\hat{D}(k)$ in the length- or average delay-based burst assembly algorithm is less than t_0, the estimate $\hat{D}(k)$ carried by the BHP in the signalling protocol is replaced by $\max(\hat{D}(k), t_0)$. The total burstification and pre-transmission delay when a fast reservation (FR) TAG protocol is used is

$$T_{FR} = \max(t_0, D + t_x) = \max(t_0, \hat{D} - \varepsilon) \,, \tag{18}$$

for the T$_{MAX}$ algorithm, $T_{FR} = \max(T_{MAX}, t_0)$], while if a standard reservation TAG protocol is used, it is

$$T_{SR} = D + t_0 \,. \tag{19}$$

Comparing these two expressions, the delay reduction achieved through pipelining by the proposed FR protocol becomes evident.

It is natural to assume that the parameters T_{MAX}, L_{MAX} and T_{AVE} in the corresponding burst assembly algorithms, and of the parameters N_{MIN}, α, T_A, L_{MIN} in the burst assembly schemes proposed in Section 3.2.4.1, are chosen so that the average burst assembly duration satisfies $E(D) > t_0$. For example, in the time-based algorithm, it is natural to choose $T_{MAX} > t_{0,}$ since otherwise we could extent the burst assembly period to get larger bursts without any cost in delay. So under these assumptions we can see that the amount of time by which the total end-to-end delay is reduced using an FR protocol, is approximately equal to the time offset $t_{0.}$

Choice of the safety margins δ and ε

The safety margins *δ* and *ε* mentioned earlier are used to reduce the effects of prediction errors in the proposed fast reservation (FR) schemes. Specifically, we add a small margin *δ* in the estimated burst length $\hat{L}(k)$, in order to reduce the probability that bandwidth is reserved for less time than the actual burst duration. We also subtract a small margin *ε* from the estimated burst assembly duration $\hat{D}(k)$, since $\hat{D}(k)$ is used to calculate the starting times of the reservations at intermediate links, and in case of uncertainty, it is safer to start reservations a little earlier than the predicted time, in order to be reasonably certain that the burst will find capacity already reserved for it when it arrives at a node.

The values of *δ* and *ε* significantly impact the success probability of the BHP reservation (the larger *δ* and *ε* are, the larger the probability) and the system costs (the smaller *δ* and *ε* are, the smaller the time interval during which capacity is reserved but not used). To obtain a good success probability without substantially increasing system costs, *δ* is chosen to be a multiple of the root mean square (RMS) of the sample residuals of the LPF,

$$\delta = c_\delta \cdot \sqrt{\frac{\sum_{i=1}^{N} e_L^2(k-i+1)}{N}}, \qquad (20)$$

where c_δ is a small constant (e.g., 2 or 3), to be referred to as the burst length correction parameter in the rest of the document, and $e_L(k)$ is the residual error between the actual and the predicted burst length. Similarly, *ε* is calculated using the corresponding RMS of the residual errors $e_D(k)$ between the actual and the predicted burst assembly durations, according to

$$\varepsilon = c_\varepsilon \cdot \sqrt{\frac{\sum_{i=1}^{N} e_D^2(k-i+1)}{N}}, \qquad (21)$$

where c_ε is the duration correction parameter constant.

3.2.4.3 *Linear Predictor LMS*

The Least Mean Square (LMS) filter [Liu03] has been chosen as the linear predictor in our work. This predictor, also used in [Ostring01], [Morato01], is simple, fast and effective, and has small computational overhead.

The estimate $\hat{N}(n)$ of the number of packet arrivals during the n^{th} frame is obtained as

$$\hat{N}(n) = \sum_{i=1}^{h} w_N(i) \cdot N(n-i), \tag{22}$$

where $N(n-i)$ is the number of packet arrivals during the $(n - i)^{th}$ frame and h is the length of the filter.. The estimate $\hat{L}(n)$ of the length of the n^{th} burst is similarly obtained as

$$\hat{L}(n) = \sum_{i=1}^{h} w_L(i) \cdot L(n-i)\,, \tag{23}$$

where $L(n-i)$ is the length of burst $n - i$. Finally, the estimate of the n^{th} burst assembly duration $\hat{D}(n)$ is obtained as

$$\hat{D}(n) = \sum_{i=1}^{h} w_D(i) \cdot D(n-i)\,, \tag{24}$$

where $D(n-i)$ is the duration of the $(n - i)^{th}$ burst.

There are a variety of ways to obtain the filter coefficients $w(i)$, i=1,2,…,h. In our experiments we used the LMS-based recursive LPF that updates the filter coefficients using a simple and efficient algorithm. Specifically, the coefficients for the k^{th} prediction period are obtained according to:

$$w_N(k) = w_N(k\text{-}1) + \mu \cdot e_N(k-1) \cdot N(k\text{-}i)\,,$$

$$w_L(k) = w_L(k\text{-}1) + \mu \cdot e_L(k-1) \cdot L(k\text{-}i)\,,$$

$$w_D(k) = w_D(k\text{-}1) + \mu \cdot e_D(k-1) \cdot D(k\text{-}i)\,,$$

where μ are adjustable parameters (steps), $e_N(k-i)$ is the error between the actual and the predicted number of packet arrivals during the $(k - 1)^{th}$ frame, $e_L(k-i)$ is the error between the actual and the predicted length of the $(k - 1)^{th}$ burst, and $e_D(k-i)$ is the error between the actual and the predicted duration of the $(k - 1)^{th}$ burst assembly period. The time complexity for the coefficient calculation of the LMS-based approach is $O(N)$.

3.2.4.4 *Performance Analysis and Simulation Results*

In this section we present simulation results on the performance of the proposed burst aggregation schemes and fast reservation (FR) protocols.

Burst Assembly Techniques

Using the Matlab environment, we simulated the burst aggregation process at an ingress queue in order to evaluate the performance of the N_{MIN}, *α*, T_A, and L_{MIN} schemes proposed in Section 3.2.4.1, and compare it to that of the previously proposed T_{AVE}, T_{MAX}, L_{MAX} schemes. We also quantified the impact the choice of the parameters N_{MIN}, *α*, T_A and L_{MIN} and of the frame size *τ* and filter order *h* have on performance.

It is useful to remind the reader that each of the proposed schemes corresponds to a different Burst Size versus Packet Burstification Delay curve (see the discussion in Section 3.2.4.1 and Figure 32), while the choice of the parameters involved (N_{MIN}, *α*, T_A, L_{MIN}, T_{AVE}, T_{MAX}, L_{MAX}) determines the exact points on each curve the burst assembly process is operating at, that is, the desired trade-off between burstification delay and burst size.

Simulation Parameters

In our experiments, the arrivals at the ingress queue were obtained from an Exponential-Pareto traffic generating source of rate *r* bits/sec. The traffic source generates superpackets (they can also be viewed as busy periods) with exponentially distributed interarrival times of mean 1/*λ* seconds. The size of each superpacket follows the Pareto distribution with shape parameter *β*. If a super-packet has size greater than *l* bytes, which is taken to be the size of the packets used in the network, it is split and sent as a sequence of packets of size *l*. The time units used for displaying our results are measured in packet slots, where 1 slot = *l*/*r* (the transmission time of a packet).

The values of the parameters used in our experiments were *β* = 1.2, *r* = 1Gbps, *l* = 1500 bytes, and *1*/*μ* = 60KB. We used 1/*λ* = 1.6 msec or 4.8 msec, corresponding to load utilization factors *p* = 0.1 and *p* = 0.3. The parameter *β* determines the Hurst parameter *H* = (3 - *β*) / 2, which takes values in the interval [0.5, 1) and defines the burstiness of the traffic. The closer the value of *H* is to 1, the more bursty is the traffic generated.

Predictor Performance

The accuracy of the estimations produced by the LMS predictor used in the burst assembly schemes of Section 3.2.4.1 can be assessed by the relative error of the prediction, defined as the inverse of the signal-to-noise ratio:

$$SNR^{-1} = \frac{\sum e^2(k)}{\sum N^2(k)},$$

where $N(k)$ is the actual number of packet arrivals during the k^{th} frame, $\hat{N}(k)$ is its estimated value at the beginning of that frame, and $e(k) = N(k) - \hat{N}(k)$. The results of Figure 37 examine the dependence of the performance of the LMS predictor on the frame duration *τ*, the order of the prediction filter *h*, and the traffic load *p*. In particular, Figure 37a shows the way the relative error varies with the frame duration *τ* for bursty traffic (*H* = 0.9). As expected, short frame durations result in smaller values of relative error, since for bursty traffic, the traffic characteristics remain static only for short periods of time. For light traffic, the predictor's performance is worse than it is for heavy traffic. This can also be seen in Figure 37b, which illustrates the impact the order of the filter has on relative error. This figure also demonstrates that there is very little improvement when the order of the filter is increased beyond a certain value. This is in agreement with the results in [Xu03], where it was argued that the performance of linear predictors for internet traffic is dominated by short-term correlations, and we don't have to “look deep" into the history of traffic arrivals to obtain a valid estimation. A small order of filter is, therefore, preferable, since it also implies smaller computation overhead. As the frame size *τ* increases, the relative prediction error remains steady after a certain value (*τ* > 0.005sec) when the traffic is light (*p* = 0.1, *p* = 0.03), while it worsens slightly for heavier traffic (*p* = 0.3).

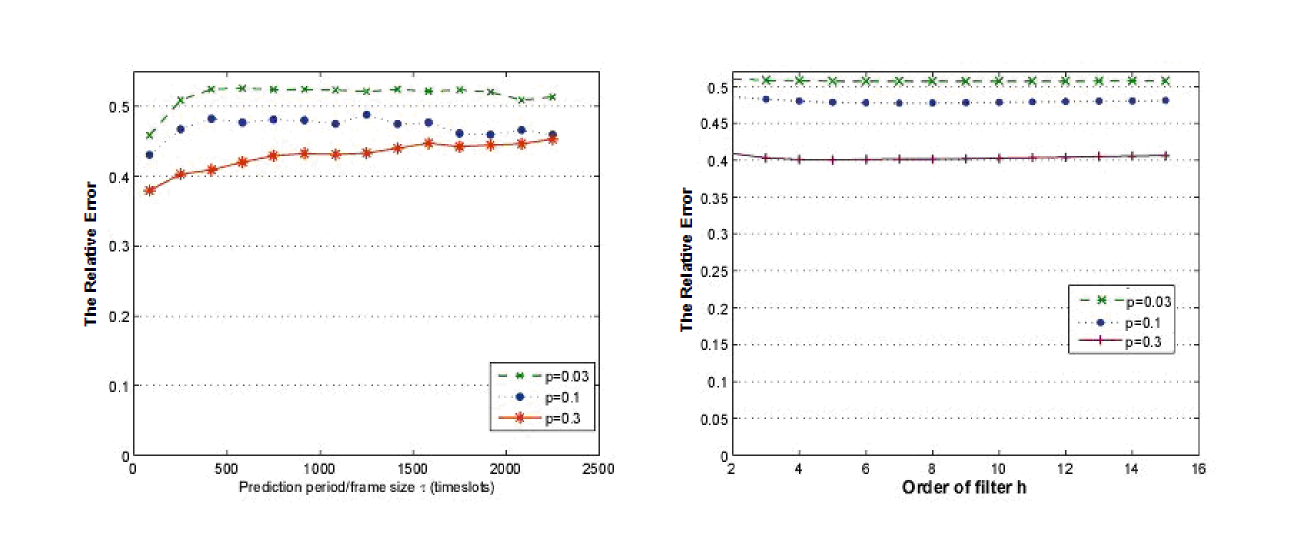


Figure 37: LMS performance for different loads p and various values of: (a) the prediction period *τ*, (b) the length *h* of the predictor.

Comparison between the Burst Assembly Schemes

In this section we compare the average burst size versus average burstification delay performance of the proposed burst assembly schemes to that of the previously proposed T_{AVE}, T_{MAX}, L_{MAX} schemes. The results reported here were obtained for bursty traffic (*H* = 0.9) and varying load utilization factor *p*. The length *h* of the LMS predictor was set to 4, while the frame size *τ* varied depending on the traffic load. The parameters of all the schemes were chosen to produce average burstification delays that lie in the same range so that the resulting burst sizes can be compared. Time delays are measured in slots. Figure 38a, Figure 39a and Figure 40a illustrate the average burst size versus the average packet burstification delay when the traffic load is *p* = 0.03, 0.1 and 0.3, respectively. The labels in Figure 38b, Figure 39b and Figure 40b display the details on the values of the parameters N_{MIN}, T_A, L_{MIN} and *α* that give the corresponding results.

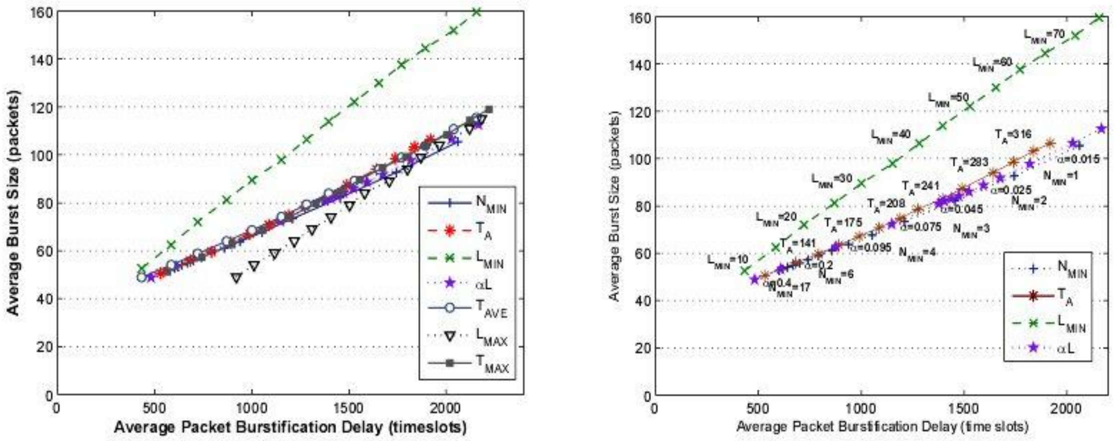


Figure 38: Performance of the proposed algorithms for traffic load *p* = 0.03: (a) Comparison of the proposed schemes with previously proposed algorithms. (b) Details on the values of the parameters used in the proposed algorithms.

Figure 38a, Figure 39a and Figure 40a show that the L_{MAX} algorithm exhibits (as expected) the worst performance for light load (*p* = 0.03), while its performance becomes relatively better for heavier load (*p* = 0.1 and 0.3). The opposite is true for the N_{MIN} algorithm, whose relative performance is worse for heavy traffic, and improves for light traffic. For a given traffic load, the N_{MIN} algorithm exhibits worse relative performance when the parameter N_{MIN} is set at low values so as to produce large bursts. This is because for small values of N_{MIN}, the algorithm cannot well tolerate estimation errors. The *αL* algorithm always performs better than the N_{MIN} algorithm, but does not succeed in outperforming some of the other algorithms considered. For a given traffic load, its relative performance compared to the other algorithms does not change with the choice of the parameter *α* (small values of *α* produce longer bursts as it can be seen in the figures). Among the previous burst assembly schemes (T_{MAX}, L_{MAX}, T_{AVE}), the T_{AVE} algorithm gives the best performance. The proposed T_A algorithm outperforms the T_{AVE} algorithm, but the improvement is rather small, as shown in Figure 40a. The improvement is more pronounced when the T_A algorithm generates longer bursts and when the traffic load is heavier.

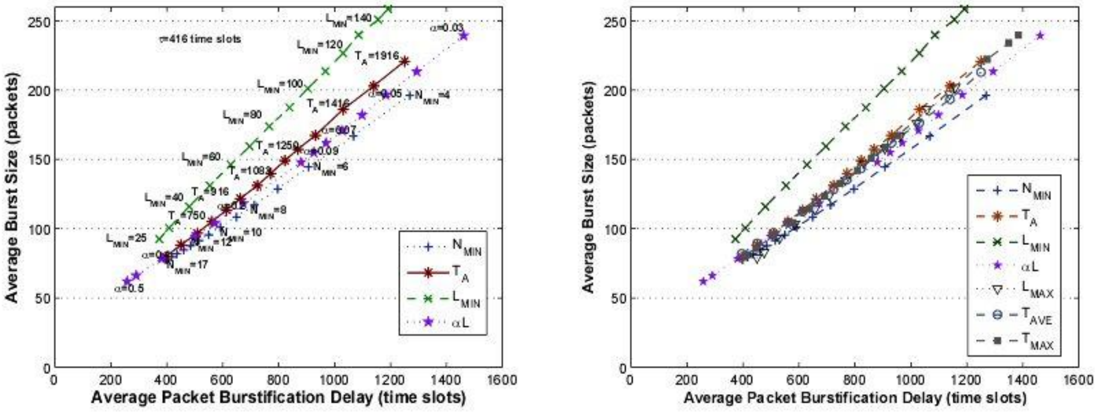


Figure 39: Performance of the proposed algorithms for traffic load *p* = 0.1: (a) Comparison of the proposed schemes with previously proposed algorithms. (b) The parameters applied on the proposed algorithms.

The best performance is consistently demonstrated by the L_{MIN} algorithm, which achieves a 33% improvement over the T_A algorithm (the second best) for light traffic load (*p* = 0.003 and 0.1) and an 8% improvement for heavier traffic load (*p* = 0.3). For a given average packet burstification delay, the L_{MIN} algorithm produces bursts of larger average size than all the other algorithms considered. The L_{MIN} algorithm can be considered a variation of the L_{MAX} algorithm, enhanced with the ability to predict the time periods where the value of DBR is expected to improve because of a large number of future packet arrivals. Note that in most of the figures, the curve that corresponds to the L_{MIN} algorithm is parallel to and above that of the L_{MAX} algorithm.

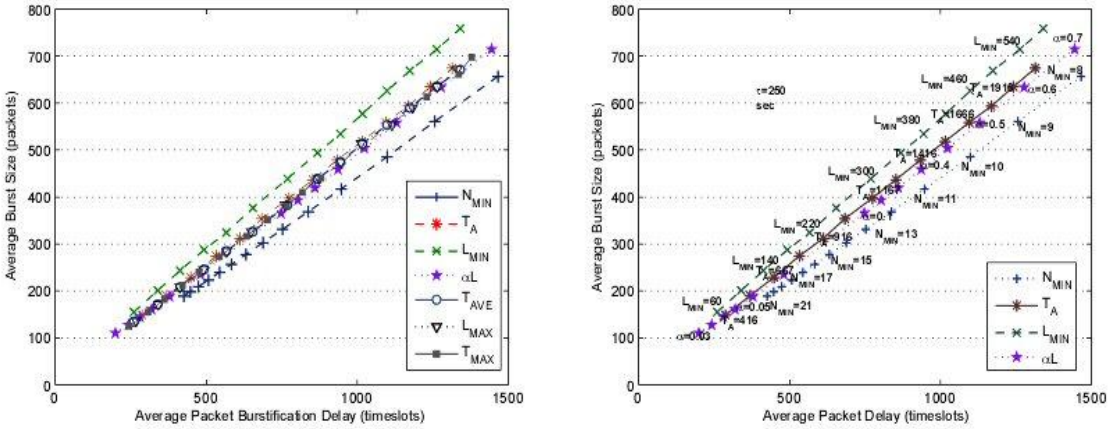


Figure 40: Performance of the proposed algorithms for traffic load *p* = 0.3: (a) Comparison of the proposed schemes with previously proposed algorithms. (b) The parameters applied on the proposed algorithms.

Fast Reservation Protocols

In this section we evaluate the performance of the fast reservation (FR) protocol of Section 3.2.4.2, using an OBS network simulator [ns2-OBS] based on the ns-2 platform [ns2]. The FR protocol was combined with the T_{MAX}, L_{MAX}, and T_{AVE} burst assembly schemes, and used predictions to estimate the corresponding burst length and/or the burst assembly durations.

In our experiments we use a simple OBS network consisting of two edge (ingress and egress) nodes and one core node. A link's bandwidth per channel is equal to 10 Gbps. The arrivals at the ingress node were obtained from a Pareto traffic source with rate *r* = 1Gbps, while the mean On and Off periods were equal to 0.002 msec and 0.001 msec respectively. The traffic source generates packets of fixed size, equal to *l* = 1500 bytes. In each experiment we change the shape parameter *β* that determines the Hurst parameter *H* = (3 - *β*) / 2 and defines the burstiness of the traffic. Specifically, we used the values *β* = 1.2, 1.4, 1.6, 1.8; the closer the value of *H* is to 1, the burstier the traffic can be characterized. Also, we use a 16 order LMS filter, even though a filter with a smaller order could also have been used.

For the T_{MAX} assembly algorithm we used the following values for the parameter T_{MAX}: 0.006 sec, 0.008 sec, 0.01 sec. For comparison purposes, the corresponding values for the parameter L_{MAX} of the L_{MAX} assembly algorithm were 488 KB, 651 KB, 813 KB, and were calculated based on:

$$L = \rho \cdot R \cdot D, \tag{25}$$

where *D* is the average burst assembly duration (T_{MAX} parameter), *L* the average burst lengths (L_{MAX} parameter) and *R* the rate of the Pareto traffic. The traffic load *ρ* is defined as

$$\rho = \frac{on_{period}}{on_{period} + off_{period}}. \tag{26}$$

Finally, for the average delay–based assembly algorithm the values of the parameter T_{AVE} we used were one half of the corresponding values used for the parameter T_{MAX} (that is, 0.003 sec, 0.004 sec, and 0.005 sec).

In our experiments we measured the following performance metrics:

- The relative error of the prediction of the burst size and the burst assembly duration, defined as the inverses of the signal-to-noise ratios

$$SNR^{-1} = \frac{\sum e_L{}^2(k)}{\sum L^2(k)},$$

$$\text{and} \quad SNR^{-1} = \frac{\sum e_D{}^2(k)}{\sum D^2(k)},$$

respectively, where $e_L(k) = L(k) - \hat{L}(k)$ and $e_D(k) = D(k) - \hat{D}(k)$.

- The probability of the BHP performing a successful reservation. In the T_{MAX} algorithm a reservation is successful if the size of the predicted burst (plus δ) is bigger than the actual burst size, that is, if Eq. (16) holds. In the L_{MAX} algorithm a reservation is successful if the predicted burst assembly duration (plus ε) is larger than the actual assembly duration, that is, if Eq. (15) holds. Finally in the T_{AVE} algorithm a reservation is considered successful if Eq. (13a) and Eq. (13b) are both valid. Note that a failed reservation does not result in a burst loss, since the estimates are compared to the real values, when the burst assembly is completed, and a new BHP is sent if needed. It only results in some loss of efficiency in the small interval between the reservation and the cancellation, where the resource remains idle.

Generally we want to have small relative errors and a large probability of successful reservations.

We first present the results for the FR protocol when combined with the T_{MAX} and L_{MAX} burst assembly algorithms in order to separately evaluate the two LMS filters used. Subsequently, we present the results obtained for the FR protocol when combined with the T_{AVE} algorithm, where both LMS filters are used.

FR protocol combined with the T_{MAX} algorithm

Figure 41 illustrates the relative error of the burst length prediction versus the shape parameter β, for different values of the burst assembly duration T_{MAX}. We observe that the relative error of the burst length prediction is quite small and decreases as the shape parameter β increases.

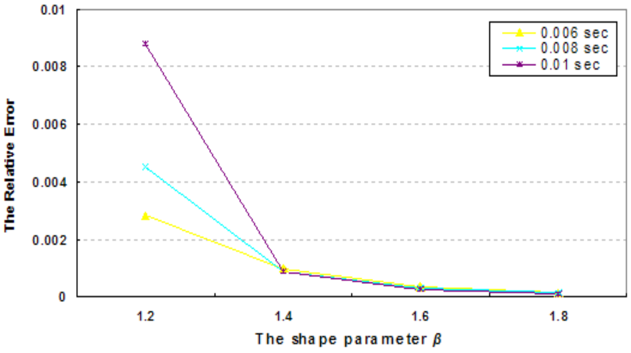


Figure 41: The relative error of the burst length prediction versus the shape parameter β, for three different values of the burst assembly duration $D = T_{MAX}$ for the T_{MAX} assembly algorithm.

Figure 42 illustrates the cumulative distribution function (cdf) of the errors of the burst length filter predictions for a burst assembly period $D = T_{MAX} = 0.01$ sec (which corresponds to about 813 KB of burst length) and shape parameter $\beta = 1.4$. The average error measured is equal to 0.135831 KB, which is quite small compared to the average 813 KB burst length, and is due to statistical fluctuations (the estimator is unbiased).

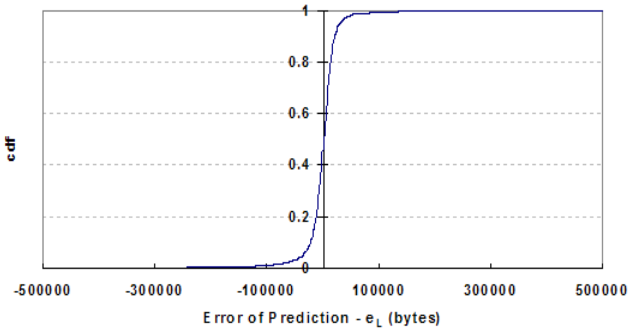


Figure 42: The empirical cumulative distribution function (cdf) of occurrences of the burst length prediction errors e_L, for the T_{MAX} assembly algorithm and assembly duration $T_{MAX} = 0.01$ sec.

The probability of successful reservations was also evaluated for safety margins δ that correspond to different correction parameters $c_\delta = 0, 1, 2, 3$. The experiments were conducted for burst assembly period $D = T_{MAX} = 0.006$ sec. Figure 43 shows the probability of the BHP successfully performing a reservation for various values of the shape parameter β and correction parameter c_δ. We observe that the results obtained for c_δ equal to 2 or

3 are quite satisfactory, and the probability of successful reservations increases as c_δ increases. These results are consistent with the results also presented in [Morato01].

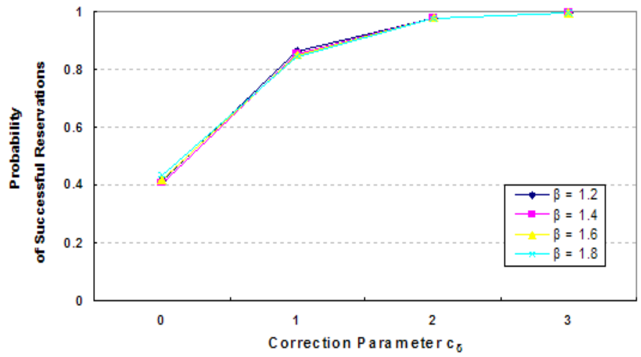


Figure 43: The probability of successful reservations versus the correction parameter c_δ, for different values of the shape parameter β for the T_{MAX} assembly algorithm. The burst assembly period was T_{MAX} = 0.006 sec. A choice of 2 or 3 for the correction of the correction parameter c_δ is adequate for obtaining very satisfactory performance.

FR protocol combined with the L_{MAX} algorithm

Figure 44 shows the relative prediction error for various values of the shape parameter β and three different choices for the burst lengths L_{MAX}. These three values correspond to the values we used for the burst assembly duration *D*, according to Eq. (12). As in the case of the T_{MAX} algorithm, we observe that the relative error of the prediction decreases as the shape parameter β increases.

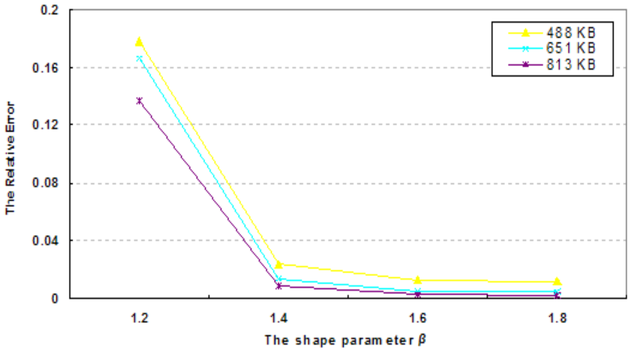


Figure 44: The relative error of the prediction versus the shape parameter β, for the L_{MAX} assembly algorithm and three different values of the burst size L_{MAX}.

Figure 45 illustrates the cumulative distribution function (cdf) of the prediction errors for $L = L_{MAX} = 813$ KB (which corresponds to about 0.01 sec burst assembly duration) and shape parameter $\beta = 1.4$. The average prediction error was equal to 0,000194 sec, which is quite small (compared to the 0.01 sec average assembly duration) and is due to statistical fluctuations. Note that the error empirical cdf is not exactly symmetric (the probability that the error is negative is 0.4), but the error is generally very small.

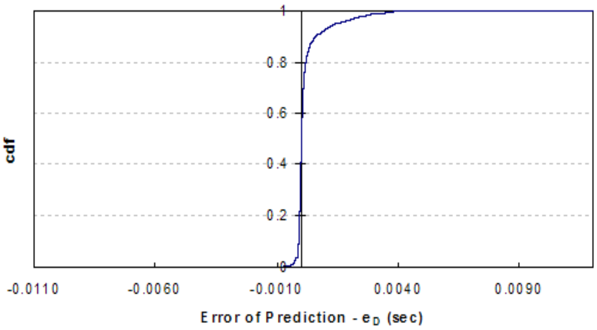


Figure 45: The cumulative distribution function (cdf) of occurrences of the errors e_D of the burst assembly duration filter predictions, for the L_{MAX} assembly algorithm and burst size $L_{MAX} = 813$ KB.

The probability of successful reservations of the L_{MAX} algorithm was evaluated for different safety margins ε, using correction parameters $c_\varepsilon =$ 1, 1.02, 1.05, 1.08, 1.1, 1.5, 1, 2, and burst size $L = L_{MAX} = 813$ KB. Figure 46 shows the probability of successful reservations for various values of the shape parameter β and correction parameter c_ε. We observe that choosing $c_\varepsilon = 1.5$ or 2 provides sufficiently high success probability and the probability of successful predictions increases as c_ε increases.

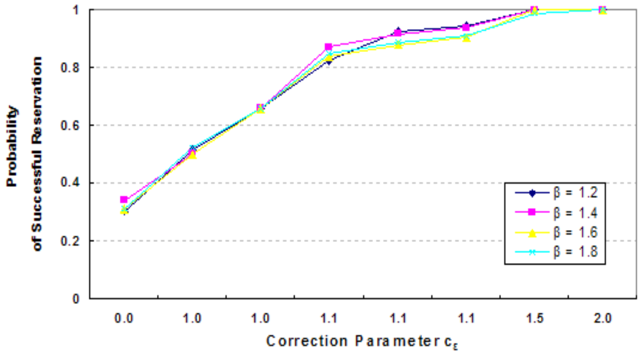


Figure 46: The probability of successful reservations versus the correction parameter cε, for four different values of the shape parameter *β*. The burst size was chosen equal to L = L_{MAX} = 813 KB. Choosing the parameter c_ε to be larger than 1.5 gives very satisfactory performance.

FR protocol combined with the T_{AVE} algorithm

When the FR protocol is combined with the T_{AVE} assembly algorithm, we have to use two LMS filters in order to predict the length of the next data burst and the burst assembly duration. Figure 47 shows the probability of successful reservations, using non-zero safety margins *δ* and *ε*; in particular, we used a burst length correction parameter $c_\delta = 3$ and an assembly duration correction parameter $c_\varepsilon = 2$. The success probability is calculated by multiplying the probability of successful predictions of the two LMS filters used. The corresponding experiments were conducted for average delay parameter T_{AVE} = 0.003 seconds. From Figure 47 we observe that the success probability is quite high and slightly deteriorates when the shape parameter *β* of the Pareto traffic increases. As already mentioned a failed reservation does not result in burst loss, but only has some (relatively small) effect on efficiency.

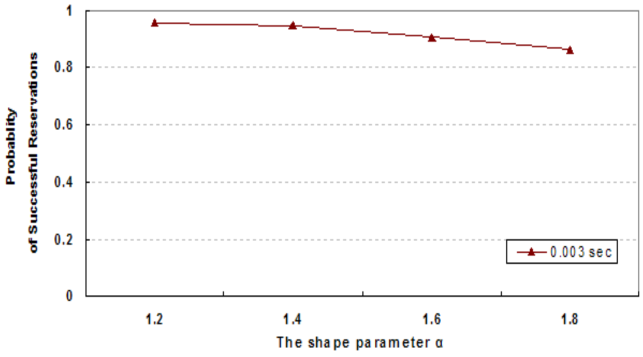


Figure 47: The probability of a BHP successful reservation versus the shape parameter β, for the T_{AVE} assembly algorithm with $T_{AVE} = 0.003$ sec. We used length correction parameter $c_{\delta} = 3$ and assembly duration correction parameter $c_{\varepsilon} = 2$.

In Figure 48 we compare T_{MAX}, L_{MAX} and T_{AVE} (for both filters) relative errors, for different values of the Pareto shape parameter β. We observe that the filters predicting burst assembly durations have larger relative errors than the filters predicting the burst sizes.

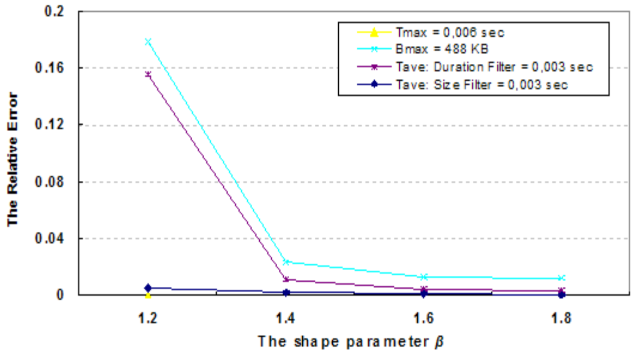


Figure 48: The relative errors of T_{MAX}, L_{MAX} and T_{AVE} versus the shape parameter β.

3.2.4.5 *Discussion*

We proposed four new burst assembly schemes for Optical Burst Switched (OBS) networks that are based on traffic prediction. The L_{MIN} assembly scheme seems to be the algorithm of choice when the average burstification delay (for a given burst size) or the average burst size (for a given burstification delay) is the criterion of interest. One should note, however, that the T_{AVE} and the T_{A} algorithms may be preferable when the delay jitter is the main consideration (both of these algorithms also give a satisfactory average burstification delay to average burst size ratio). We also described a fast reservation protocol that can be combined with the T_{MAX}, L_{MAX}, T_{AVE} or the other proposed burst assembly schemes, to further reduce the pre-transmission delay in OBS networks. This is done by sending the Burst Header Packet in the core network before burst assembly is completed at the ingress node. We find that the probability of successful reservations using fast reservations is very satisfactory, provided that a small correction term (2 to 3 times the root mean square error) is added to the predicted burst length and assembly duration estimates. The proposed schemes and protocols can be used to reduce the end-to-end delay and increase the size of the bursts produced, while making efficient use of the bandwidth, and maintaining a good probability of successful prediction for the reservations needed in an OBS network.

3.3 Hybrid Approaches

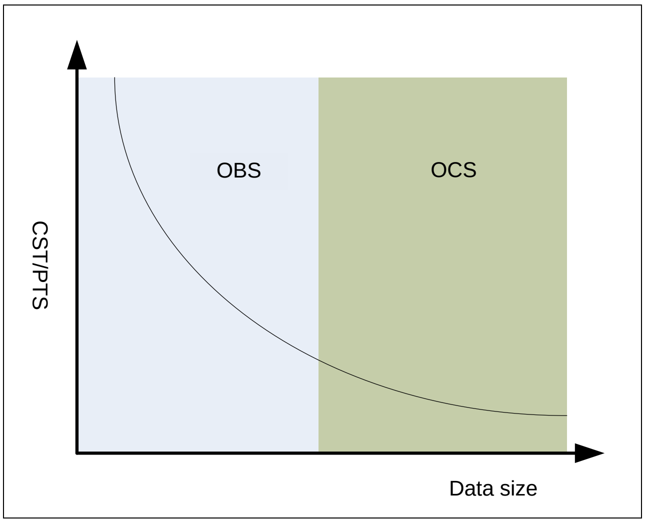


Figure 49: Signalling overhead is relatively important for small bursts

While Optical Circuit Switching (OCS) and Optical Burst Switching (OBS) are sophisticated and widely deployed (in the case of OCS) technologies in current photonic networks, they both pose some imperfections. Whereas OCS has many advantages in the transport of large data streams, it lacks in conveying small data packets to its destination. OBS on the other hand is excellent in transferring bursty traffic but is deficient in sending large and stable flows. In Figure 49, the data size of a packet is plotted against the ratio of the circuit setup signalling time (CST) and the data packet transmission time (PTS). We can conclude from this graph that both OBS and OCS have their domain where they reach optimal performance: larger data packet sizes reduce the importance of the signalling overhead.

To exploit the merits of both technologies, a hybrid approach can be used; we describe two alternatives in the following sections.

3.3.1 Parallel choice

The first hybrid technique decides at the source node which switching technology to use. The decision is based on the length of the data size before transmission: OBS for relatively small data packets, OCS for the larger and stable data streams. This decision can be backed up by the properties of both switching techniques. When a small data packet is sent using OBS, the relatively large overhead for setting up a circuit between the source and destination is not needed anymore and can be replaced by a set of small Burst Header Packet (BHP) checks to forward the data burst. We note that the use of OBS can be an overlay technique where OBS makes use of OCS connections between different OBS nodes. When a Grid application needs to stream a huge amount of data through the network using OCS, it will benefit by the onetime circuit setup signalling. After the signalling is done, the application can start to send without the need of extra signalling operating cost.

3.3.1.1 *Hybrid switch design*

This kind of hybrid switching requires a special kind of switch. The basic function of such a kind of switch is straightforward: it must forward each data packet coming from a certain in-port to the correct outgoing port. The decision which out-ports to take is made in a special control unit of the switch, which makes this decision based on the control information accompanying the data packet. In case of OCS, this is done in advance, by the circuit setup process. In the case of OBS, a special kind of packet (BCH or Burst Control Header) in send out of band before the actual data packet, which informs the control unit where the data burst must go to. The time between the BHP and the data burst is denoted as T_{offset}. Every switching fabric is restricted, due to its physical impairments, by its switching speed T_{switch}. To be able to switch successfully, the following statement must evaluate to true: $T_{switch} < T_{offset}$.

For long data transfers (circuit), slow switching speeds are usually enough to obtain a high switch utilisation. However, for small data transfers (e.g. OBS) high speed switching fabrics are required to achieve an acceptable bandwidth throughput in the switching nodes. Current switching fabrics offer a broad range of switching speeds, but there is always a penalty in terms of either speed and scalability or cost. For example, micro-electromechanical systems (MEMS) offer good scalability but have a very slow switching speed as a result of the mechanical tuning. In contrast, Semiconductor Optical Amplifiers (SOA) can only scale up to 32x32 port counts at a very high cost, but can achieve switching speeds in the nanoseconds domain. The proposed construction of such a switch would consist of two parts: a part supporting a large set of slow switching wavelengths and a part maintaining only a minimal set of very fast switching wavelengths. This way, we can combine the best of both worlds, using only a subset of the spectrum for fast switching traffic and the rest for slow switching traffic.

3.3.1.2 *Wavelength scheduling*

The suggested algorithm to send data in a hybrid fashion is depicted in Figure 50. At the edge of the network, the size of the data to be sent is calculated. If this size oversteps a boundary threshold or the application sending the data knows it is a streaming application, OCS is chosen as switching technique on a slow switching wavelength. In the other case, OBS is advised as switching technique. In the case of OBS, the following step includes checking if there still is a fast switching wavelength available. If so, this wavelength is chosen and the data transfer can begin. In the other case, a slow available wavelength is chosen with a large enough T_{offset} for the burst so that the $T_{switch} < T_{offset}$ still can evaluate to true for other switches in the network.

This algorithm has a flaw. OCS will never use a fast switching wavelength in order to preserve the fast switching bandwidth for the bursty traffic. In case the network does not have to handle a lot of bursty traffic, this (relative small) portion of bandwidth remains unoccupied. In the other case where the network has to attend a lot of bursty traffic these wavelengths will be utilized in an optimal way.

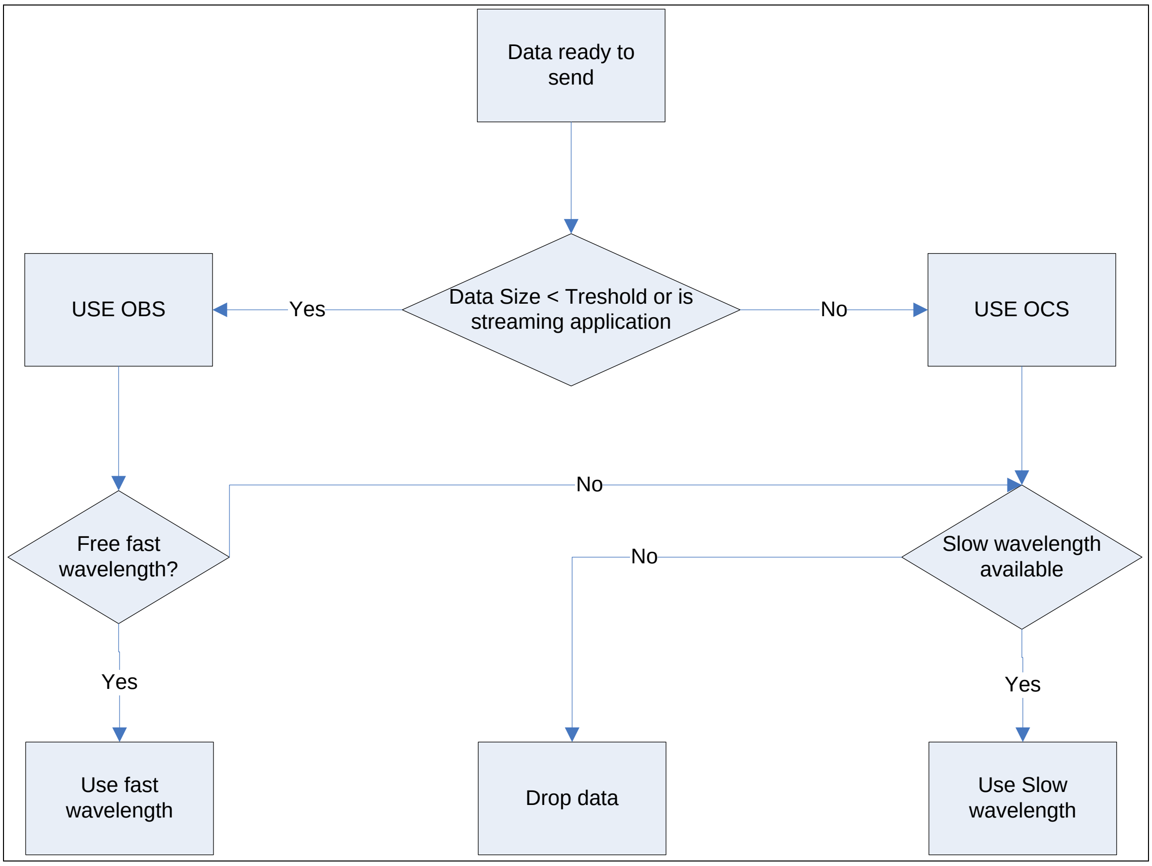


Figure 50: The proposed flowchart for sending data at the edge of the network

3.3.2 Burst-over-Circuit-Switching

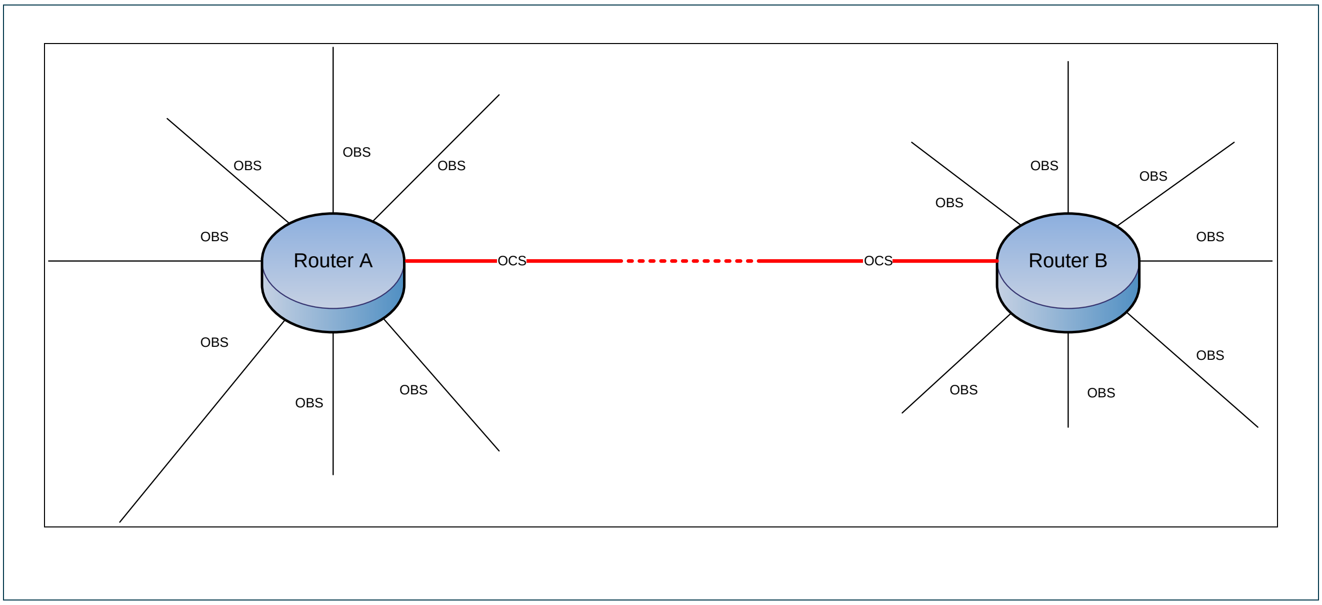


Figure 51: Sending OBS traffic over an already setup circuit.

Another way to combine both approaches could be to strategically construct circuits on links with a high traffic throughput. Offloading traffic to direct light paths and bypassing intermediate OBS nodes reduces the number of burst-mode capable switch interfaces in the network Also, it avoids control processing and contention situations.

This is illustrated in Figure 51. The Router A receives packets from the incoming OBS links that have to go to Router B. Now instead of checking all the BHP's and forwarding these packets to the next OBS router, these OBP's can be stuffed into the OCS circuit. This circuit has been reserved in advance and will eventually lead to Router B without intermediate hops having to test out the BHP's. These kind of circuits can be cascaded: from hop 1 through a OCS tunnel to hop 2 (which is a hybrid OCS/OBS router), where it is decided that this OBS message once again will be forwarded through another OCS tunnel and so forth…

We can look at this class of hybrid switching via the client-server principle. The client layer is an OBS network and the server layer is a wavelength-switching network. OBS or OPS nodes mostly aggregate traffic at the edge of the core network. These nodes are interconnected across the core network by direct light paths in the underlying wavelength-switched network. Optical bursts/packets are only switched in the client layer nodes and transparently flow in light paths through the circuit-switched server layer nodes. If the client layer nodes do not switch transit traffic, we term the approach "Burst-over-Circuit-Switching" (BoCS).

3.3.2.1 *Pre-planned versus Dynamic circuit setup*

A distinction can be made on the moment of circuit setup. In the pre-planned option, some entity in the Grid decides in advance where to put this circuit. In the dynamic option these circuits are not planned. When it is noticed that on a certain set of links, a lot of traffic is sent and a threshold has been reached, a circuit can be established on the fly between the ingress and egress point of this set of links. The pre-planned option allows the network administrator to anticipate on traffic problems that might happen in the bottleneck section of the network. On the other hand, the pre-planned option lacks flexibility for unaccounted traffic scenarios. The dynamic option will try to put circuits on places for unaccounted traffic scenarios as well.

3.3.3 Integrated Hybrid Optical networks

In this category of hybrid optical networks, OBS and OCS are completely integrated. This means that all network technologies share the same bandwidth in the same network simultaneously. Traffic is either transported in wavelength-switched or in burst-switched mode.

Each node can choose to do two things.

1. Opt to use a given wavelengths segment as part of a predetermined wavelengths path and send the traffic wavelength switched.

2. Ignore the established circuits and pass the traffic to a neighbouring node.

Choosing between the two views is done on the fly and possibly on a packet to packet basis. Choosing not to use the created circuit can for example be done in case of congestion. The node can then dynamically choose to transfer the data packets using another route using OBS. Alternatively, the choice between the two modes, can also be motivated by QoS differentiation, e.g., wavelength-switched for high priority traffic.

3.3.4 Hybrid Network Design in Phosphorus

In the previous sections, we have motivated that multi-granular (MG) switching supports various application and QoS requirements on a common transport network infrastructure. It has been demonstrated in literature [DeLeenheer08] that a multi-granular optical cross connect (MG-OXC) offers improved blocking performance for even a small number of fast ports. In the following, we will show that multi-granular switching also provides economic advantages on the network level. For this, a model for dimensioning a multi-granular optical network will be proposed, and results are obtained that illustrate the possible reductions in total network cost and improvements with regard to node scalability.

3.3.4.1 *Problem Statement*

Assume the network is composed of OXCs, capable of switching circuits or slow bursts on millisecond scale (slow MEMS switch), and fast bursts or packets on a nanosecond scale (fast SOA switch). Likewise, traffic is generated by clients requiring both fast and slow switching. The question arises how to dimension the network, given a static traffic demand with a given fraction of fast and slow traffic. The main objective is to minimize the network's cost, given a price ratio of slow over fast port costs. Another objective is to reduce the cost of the cross-connect with the highest cost, and as such obtain reduced node complexity.

The problem can be simplified by observing the very high cost and difficult scalability of fast switching fabrics [BenYoo06, Papadimitr03]. To minimize the use of fast ports, slow traffic will be switched exclusively by slow switches, and thus this minimum cost network flow problem can be solved independently with known algorithms. We do not consider this problem, and as such only need to plan the network for the remaining fast traffic. This fast traffic can be switched in one of the following ways: either on a fast switch, which can be shared between different demands[1], or on a slow switch that is then exclusively reserved for that particular demand.

We assumed no wavelength conversion is possible in the switches. Note that the proposed Integer Linear Programming (ILP) model [Nemhauser99] does not incorporate wavelength assignment, mainly because of complexity issues. Thus, in principle this model assumes full wavelength switching is available in each OXC, which has consequences for the economics of the obtained solutions [Subraman96]. Otherwise, an additional wavelength assignment step is required; an overview of optimal and heuristic approaches to this problem can be found in [Choi98, Alanyali01, Shiraz22]. We now proceed to the actual model, which has been formulated as a Linear Integer Programming model. The appeal of a linear model is that it allows the use of general-purpose techniques (simplex and interior-point methods) to find optimal solutions.

3.3.4.2 *Linear Model*

The following notations are introduced:

- directed graph G(V, E), V the set of nodes, and E the set of directed links
- each wavelength has a fixed bandwidth B, identical for all links and wavelengths
- Λ^{sd} is the fixed demand (fraction of bandwidth B) between source s and destination d
- C represents the cost ratio of fast over slow switches
- potential paths between source s and destination d are determined in advance, and are represented by the boolean parameters $\pi_{pl}^{sd} = 1$ iff link l is part of path p between source s and destination d, 0 otherwise.

The following binary variables are introduced to determine the path p to use for demand Λ^{sd}, and whether to use slow or fast switching:

- $\delta_p^{sd} = 1$ demand (s,d) uses slow switching on path p, 0 otherwise
- $\varepsilon_p^{sd} = 1$ iff demand (s,d) uses fast switching on path p, 0 otherwise

The integer variables x_l and y_l represent the number of slow and fast switching wavelengths on link l, and are given by:

$$\forall (s,d), p : x_p^{sd} \geq \Lambda^{sd} \delta_p^{sd}$$

$$\forall l : x_l = \sum_{sd} \sum_{p} \pi_{pl}^{sd} x_p^{sd}$$

[1] Sharing bandwidth between different demands is usually referred to as *grooming*.

$$\forall l : y_l \geq \sum_{sd} \sum_{p} \pi_{pl}^{sd} \Lambda^{sd} \varepsilon_p^{sd}$$

The auxiliary variables x_p^{sd} (integer-valued) represent the number of wavelengths required to carry demand Λ^{sd}. Clearly, slow switching corresponds to reserving end-to-end circuits that are exclusively accessed by the source and destination, while fast switching allows grooming of traffic on a link-by-link basis. The following constraints enforce two requirements: (a) each demand can only use a single path, thereby excluding solutions based on multi-path routing, and (b) a demand is either switched slow or fast, but not both.

$$\forall (s,d) : \sum_{sd} \left(x_p^{sd} + \varepsilon_p^{sd} \right) = 1$$

The final step to obtain total network cost is to transform the variables for the wavelength count on each link l = (u, v) (u and v represent nodes), into ports counts for each node n. For an OXC architecture in which the slow and fast switching fabrics are placed in parallel (refer to the following section for sequentially placed slow and fast fabrics), the slow and fast port counts are given by:

$$\forall n : x_n = \sum_{m} \left(x_{(m,n)} + x_{(n,m)} \right)$$

$$\forall n : y_n = \sum_{m} \left(y_{(m,n)} + y_{(n,m)} \right)$$

The first objective we propose is to minimize the total installation cost of the network, which in large part depends on the total number of installed switching ports:

$$\min \sum_{n} \left(x_n + C \cdot y_n \right)$$

Equation 8: Objective function to minimize the total installation cost of the network

A related objective function is to minimize the cost of the most expensive cross-connect. This objective is motivated by the limited scalability of OXC designs, especially when based on fast switching fabrics. This objective can be stated more formally as:

min z where

$$\forall n : z \geq x_n + C \cdot y_n$$

Equation 9: Objective function to minimize the cost of the most expensive cross-connect.

OXC Architectures

In this section, we demonstrate how the proposed model can be adapted to support the different OXC architectures that were presented in [Zervas08]. More precisely, we show how slow only, fast only and the MG-OXC alternatives (parallel vs. sequential) can be incorporated in the model.

First of all, note that the model captures two Integer Linear Programming (ILP) problems, associated with scenarios in which either only slow or only fast switching is used. Indeed, in case $\forall(s,d),p:\varepsilon_p^{sd}=0$, all demands will be served by a slow only connection (i.e. $y_l = 0$). Likewise, in case $\forall(s,d),p:\delta_p^{sd}=0$, only fast ports will be used ($x_l = 0$).

Furthermore, observe that in case slow only switching is used, the objective function can be simplified to:

$$\sum_n \left(x_n + C\cdot y_n\right) = \sum_n x_n = \sum_n\sum_m\sum_{sd}\sum_p \left[\Lambda^{sd}\left(\pi_{p(n,m)}^{sd}+\pi_{p(m,n)}^{sd}\right)\right],$$

which corresponds to the use of shortest path routing for all demands. To differentiate between the parallel and sequential MG-OXC approaches, the number of slow ports in the latter case is given by:

$$\forall n: x_n^* = x_n + 2\cdot\sum_m y(m,n)$$

This corresponds to the allocation of additional slow ports for each incoming fast wavelength that is introduced in a cross-connect. In the following section, we will demonstrate that network cost is only slightly increased, as a limited number of additional slow ports suffice to allow the configurability offered by the sequential switch designs (see [Zervas08] for more details).

Complexity

Table 5 summarizes the complexity of the different ILP models. Here, N represents the number of OXC nodes, L the number of network links, D the number of demands, and P the number of paths that are considered for each demand (assumed identical for all demands). Observe that MG parallel and MG sequential have an identical complexity. The table lists the complexity when the first objective function is used. When minimizing the highest node cost, the number of variables is increased by 1, and N additional constraints are introduced.

Design	Variables	Constraints
Slow only	2DP + L + N	D(1 + P) + L + N
Multi-granular	3DP + 2L + 2N	D(1 + P) + 2L + 2N
Fast only	DP + L + N	D + L + N

Table 5: Complexity of ILP model for different OCX designs

3.3.4.3 *Evaluation*

The ILP-formulated problems were implemented and solved through the use of the ILOG CPLEX library. All OXC design approaches are evaluated, including slow only, fast only, and both multi-granular (parallel and sequential) architectures. Results are obtained for a specific scenario, defined by the Phosphorus topology depicted in Figure 52. The traffic demand matrix is fixed, and consists of uniformly generated traffic between all source-destination pairs with average $\Lambda^{sd} = 0.05$. To reduce computational complexity, we only considered the 5 shortest paths for each demand. Results show the total network cost and highest node cost, for both objective functions.

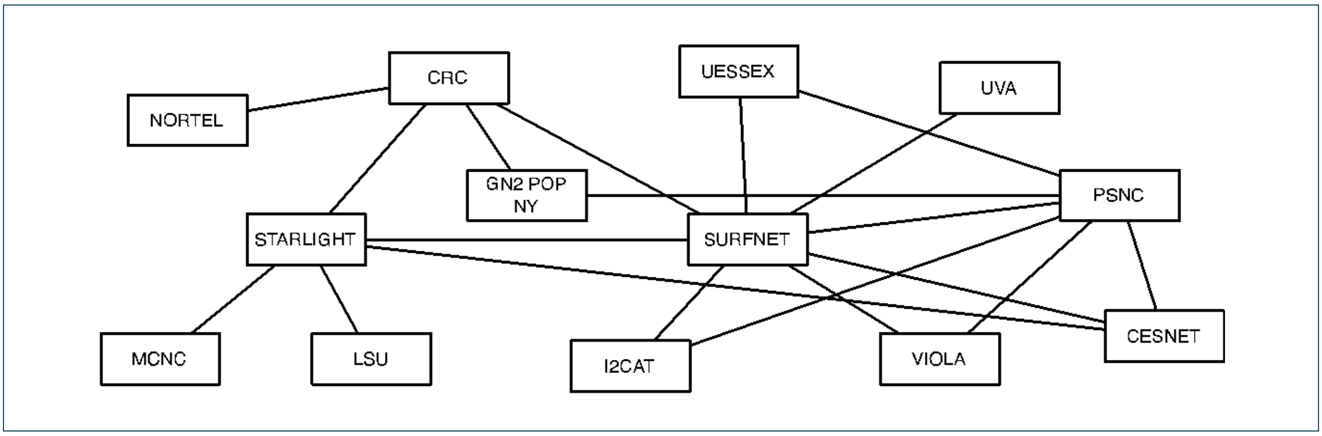


Figure 52: Phosphorus testbed topology (incl. North-American sites)

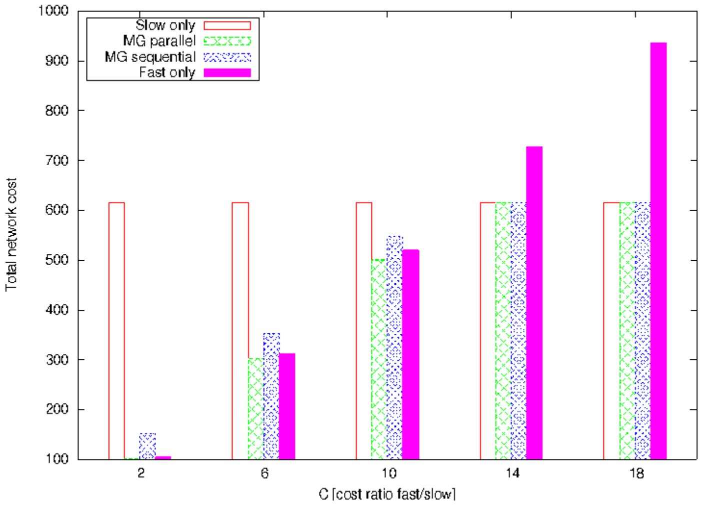


Figure 53: Minimized total network cost

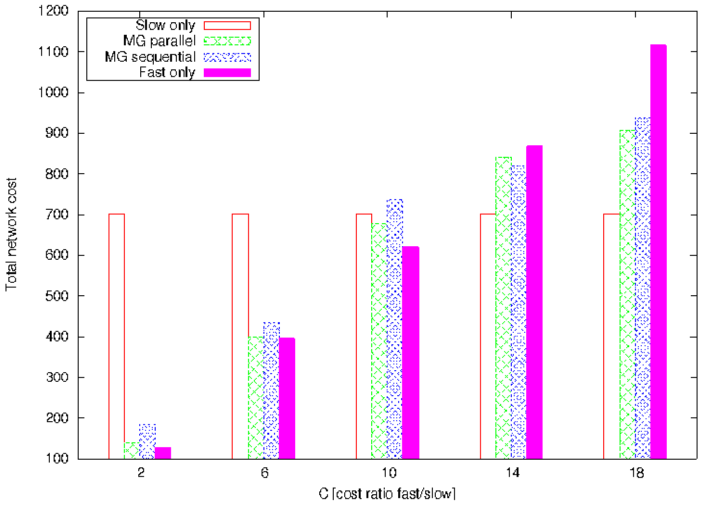


Figure 54: Total network cost for minimized largest node cost

Comparing the total network cost when minimizing either total network cost (Figure 53) or highest node cost (Figure 54), a number of interesting observations can be made. First, note that slow only returns constant network costs, due to its independence of cost ratio C. As expected, minimizing the highest node cost (**Equation 9**) slightly increases total network cost when compared to the objective defined in **Equation 8**. Furthermore, MG sequential produces total network costs at least as large as MG parallel when minimizing network cost, although this is not the case when minimizing the highest node cost. Finally, for high values of C, the multi-granular approaches return identical results as the slow only design when using objective **Equation 8**.

In summary, significant cost savings are possible when using multi-granular optical switching, in comparison to slow only or fast only switching. Also, introducing reconfigurable fast wavelengths through the MG sequential design will only slightly increase total network cost.

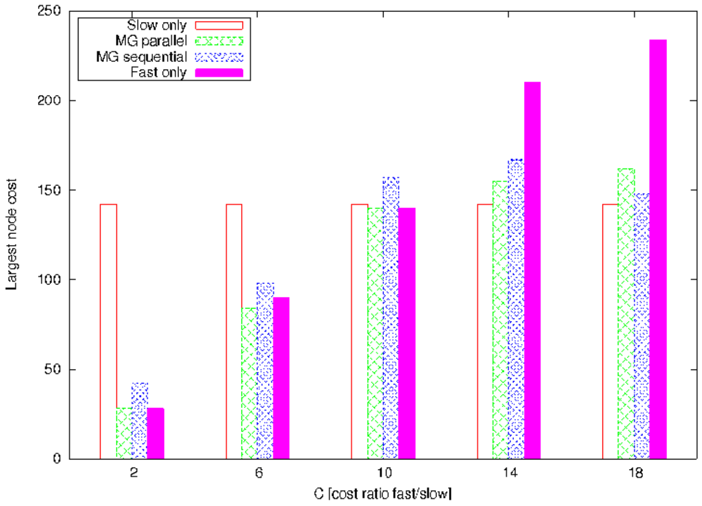


Figure 55: Minimized largest node cost

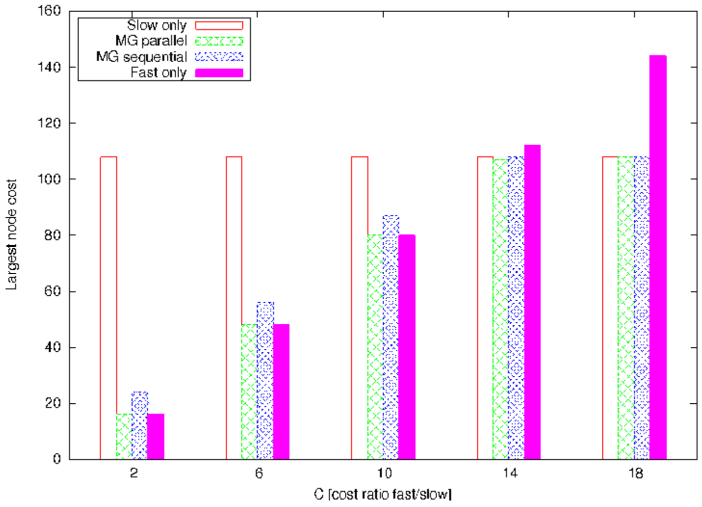


Figure 56: Largest node cost for minimized total network cost

We now consider the highest node cost when minimizing total network cost (Figure 55) or highest node cost (Figure 56). Again slow only produces constant results, but lower values are achieved by optimizing for the objective given in **Equation 9**. Observe that MG sequential returns highest node costs lower than MG parallel, only when minimizing the highest node cost. Multi-granular optical switching can thus clearly reduce the highest node cost, and consequently improve node complexity which is critical for scalability issues.

In conclusion, an ILP-based network dimensioning algorithm was introduced, and results indicated that significant cost savings can be obtained when implementing multi-granular optical switching. Furthermore, reduced node costs can be achieved as well, in order to minimize scalability problems corresponding to emerging fast switching fabrics.

4 Switching Techniques in Lambda Grids

4.1 Grid Requirements

Current Grid applications such as distributed video transcoding and multimedia rendering can generate vast amounts of traffic. Taking into account the quantity of control and resource information that Grid nodes generate, it is obvious that the network infrastructure supporting the Grid should offer a vast quantity of affordable bandwidth in a reliable way. Optical networks promise to meet the high bandwidth requirements of emerging Grid applications, since they provide transmission rates that exceed those of copper networks by several orders of magnitude at very low bit error rates. Thus, optical networks play a significant role in the evolution of the Grid as a viable and commercial product.

The development in recent years of optical devices and systems for key networking functions (e.g., amplification, regeneration, multiplexing/demultiplexing, switching, etc.) has fostered the development of wavelength-division multiplexing (WDM) as an optical circuit switching (OCS) approach. The introduction of OCS technology in the Grids (lambda Grids), has the typical advantages and disadvantages of circuit switching as opposed to packet switching networks. The utilization of the available bandwidth drops as the requirements of the individual end-users/applications decrease or have bursty characteristics. Optical Packet Switching (OPS) could be a possible solution, however the technology required for OPS is not yet mature. To this end, Optical Burst Switching was introduced to combine both strengths of packet and circuit switching. OBS allows access to bandwidth on a sub wavelength scale, and, as such, statistical multiplexing of several data transfers (called bursts) is possible on a single wavelength.

Generally, the following are the most common requirements of a user/task/application from the Grid's network resources:

- Delay is the time it takes for a packet to travel from the source (sender) to the destination (receiver). In a usual Grid scenario the user requests small or an upper bound on the end-to-end delay that his jobs and their corresponding data must face during their transmission. This is one of the most important user requirements. In order to satisfy the network delay requirements possible paths that result in large delays, may be discarded and resource reservation can be applied. In this way, the user can get an assurance that the Grid will provide the desired level of QoS. The reservation process can be immediate or undertaken in advance, and the duration of the reservation can be definite (for a defined period of time) or indefinite (from a specified start time until the completion of the application).

- Delay jitter is the variation in the delay of packets taking the same route. Delay jitter can be measured mainly in a packet switched networks, where the buffering delays and routing decisions affect the delay and possible the order in which the packets arrive at the destination. However, in a circuit switched network where an end-to-end connection is established in a form of a wavelength switched path, the packet delay jitter is negligible.

- Bandwidth is the rate at which packets are transmitted.

- Reliability of a packet transmission. A common measure of the reliability is the packet-loss rate, defined as the rate at which packets are dropped, lost, or corrupted. A Grid user might request that his connection has packet loss ratio, which is less than an upper bound.

In the traditional network, it is common that the routing policy relies on the shortest path routing. To convey data from one point to another, the shortest path in terms of the number of hops of the path is used. In a Grid scenario, this metric starts to lose its importance. The interest of the user of the Grid lies in the results of its job. Since different resources can process this job, the destination of the handling resource is of less significance. Anycast routing puts this idea to the test: users can submit their jobs with a specified service delivery without assigning a specified destination. This approach is especially useful for delivering consumer-oriented services over an optical network to a large number of users, as centralized job scheduling and Grid status monitoring on client devices can be avoided.

One way to incorporate the anycast routing principle into OBS for example, is as follows. Each resource sends out a status packet to its adjacent routers, informing the routers of its free capacity. These routers in turn forward these status packets to all neighbouring routers with an updated free capacity value proportional to the available bandwidth between the routers. This adjustment must be implemented to prevent these status updates living forever in the network. However, to prevent the looping of resource states with extremely large capacities, a status packet gets dropped after having travelled a certain number of hops. The actual routing is then done as follows. The Burst Header Packet is sent without an explicit destination before the actual data and is processed at a neighbouring router. This router checks whether it has a resource attached with sufficient capacity and delivers the burst accordingly. Otherwise, the router selects an outgoing link with the most capacity and passes the burst optically to that router without processing it. Eventually each burst will either be dropped along the way or reach a resource capable of processing the job. The use of anycast routing leads to a favourable combination of constraints for both network parameters (e.g. bandwidth, delay) and resource parameters (e.g. processing availability and speed).

4.1.1 Phosphorus Grid applications

This section presents the Grid network requirements of the Phosphorus applications and proposes a classification of them in two classes.

The KoDaVis application deals with collaborative visualizations of huge atmospheric data-sets. Large data sets with a typical size of about 1 Terabyte are pre-computed and stored at the super-computer site (not locally at the scientists' lab). A scientist-client wishing to perform a visual analysis, accesses only parts of these dataset. The collaboration among scientists is enabled through the use of a teleconference established among them. There is a parallel data-server that distributes fragments of data selected by the clients, and a collaboration server that synchronizes all clients. As KoDaVis is an interactive, collaborative application, it imposes end-to-end delay (latency) and delay jitter requirements on the network connectivity. Thus visualization data streams need to be transmitted over reliable high-bandwidth, low-latency channels to permit remote steering of the application. The time that KoDavis experiments are executed is usually pre-planed (it is similar to an online-meeting of scientists) so we can assume that the traffic introduced to the network is not actually dynamic but semi-static, in the sense that it can be known in advance that certain bandwidth will be needed at a specific time and for a specific duration.

TOPS is a visualization service which enables remote viewing of large scientific datasets (2D or 3D) on high resolution display devices (Tiled Panel Displays). The difference between KoDaVis and TOPS is that in the former the visualization is performed at the clients' sites, while in the latter a remote computation resource, closely located to the data, is used to generate the visualization, which is then forwarded to the client. Reservations are made for the graphics (computation) resources, and the network connection between the visualization service and the client (display). To this end high and on-demand bandwidth allocations are requested. Note that the bandwidth needed to support visualization applications is dependent on the resolution of the source. For TOPS, the display consists of four projectors with a resolution of 1600x1460 pixels, 24-bit color depth and a vertical refresh rate of at 105 Hz each, which requires approximately 40 Gbps (the stream is not compressed).

The DDSS (Distributed Data Storage Systems) applications are widely used to transport, exchange, share, store, backup/archive and restore data in many scientific and commercial applications. The proposed test scenarios include two DDSS use cases: (i) data transfers performed using open-source GridFTP application and (ii) backup/archive/restore operations performed by a commercial application. The communication model is one-to-one or many-to-one. The DDSS applications require medium to high bandwidth, on demand allocation of network resources and there are no latency or delay jitter issues.

The WISDOM (Wide In Silico Docking On Malaria) application is a docking workflow/service that allows the researcher to compute millions of compounds of large scale molecular dockings on targets, implicated in diseases like malaria (in silico experimentation). There is a pre-staging phase where the software and the input-data are transferred to the sites (more than one) where the docking simulation will run (estimated size of data: 0.5 GB) and a post-staging phase where the output-data (estimated size: 0.5 TB) are gathered and stored in a common data-base (many-to-one communication model). This application falls in the general category of e-science computational intensive applications. The network requirements have mainly to do with the transfer of the output data. Thus, on-demand and medium bandwidth is required, while no latency and jitter issues exist.

Table 6 shows in summary the basic requirements of the Phosphorus related applications.

Applications	End-to-end delay	Delay jitter	Reliability	Bandwidth	Traffic
Kodavis	low	low	high	High	Semi static (Known starting time, known duration, known bandwidth)
TOPS	low	low	high	high	Semi static
DDSS (Grid-FTP)	-	-	-	medium-high	Dynamic (can also be semi static)
DDSS (backup/archive/restore)	-	-	-	medium-high	Dynamic (can also be semi static)
WISDOM	-	-	-	medium	Dynamic (can also be semi static)

Table 6: Phosphorus reference applications requirements specification.

Based on these requirements two classes of Grid applications can be defined:

- Class A applications have the following characteristics: high bandwidth, low delay, low jitter, semi-static traffic, can be better served with OCS technologies. Class A applications are visualization and collaborative applications, such as KoDaVis and TOPS.

- Class B applications have the following characteristics: medium-high bandwidth requirements, on demand and dynamic bandwidth allocation, can be better served with OBS technologies. Class B are computation intensive applications such as WISDOM, which require small to medium size data transfers and data-Grid application with no interaction by the end user during the data transfer, such as DDSS.

4.1.2 OCS

Wavelength-routed OCS networks consist of different wavelengths routers which are interconnected by WDM fiber links. The basic building block of this kind of network is a light path: a consecution of hops and fiber links to provide a circuit switched connection between two end pairs.

OCS has several shortcomings, namely that it takes several requests (and therefore precious time) to setup which makes creating a circuit very undynamic. This signalling overhead is not acceptable for relatively small jobs, but is on the other hand independent of the (sometimes) large job size.

Another point is that when the circuit has finally been configured, the connection holding time equals the time needed to send the data. This way a specified QoS can usually be provided (different wavelengths are being reserved if wavelength conversion is used) which means fixed latency, no jitter, etc. Though once a circuit is setup it is mainly reliable, it is still preferable to consider protection and restoration of the resources. Several kinds of schemes exist to protect the network- and Grid resources.

1. Dedicated protection (1+1). One dedicated protection resource protects exactly one working resource, which constantly does the same work as the working resource. One could think in terms of network protection where with every light path been setup, another protection path has to be set up. Other than dedicated protection in the network, protection could also be performed on the Grid elements, e.g. by reserving two resource nodes for the execution of one job. In case one fails, the other one can still continue. A combination of resource protection in the network and Grid domain is also possible.

2. Dedicated protection with extra resource utilization (1:1). One dedicated resource protects exactly one working resource, but this protection resource can occupy itself with other work in fault-free conditions.

3. Shared recovery with extra resource utilization (1:N). A specified recovery entity is dedicated to the protection of up to N working entities. In fault free-conditions this recovery entity can be used for other purposes.

4. M:N protection. A set of M recovery entities protect a set of N working entities.

4.1.2.1 *Establishing a circuit*

The way an OCS circuit can be set up in a Grid environment, depends on the sort of application is requesting the circuit. Several models exist.

- Intelligent network: The network monitors the traffic and automatically set up light paths when high-speed data flows are being detected, sharing the network resources efficiently across applications.
- Asynchronous file transfer: the Grid application submits a request to transfer data. The network schedules and transfers the requested data from its source to its destination. The asynchrony lets the system collect the transfer requests of several applications and optimize the transfer scheduling and use of high-seed circuits. The optical circuits are created and held only as long as necessary to complete the existing transfers.

- Autonomous file transfer: the application explicitly requests the desired network connectivity and uses the dedicated network to achieve a predictable performance. The optical paths are direct, secure and do not allow sharing. This way a high QoS can be guaranteed.

Hybrid techniques

When we compare these two switching techniques together, we can conclude that there is no sole solution but that we can try to use best of both worlds. While OCS efficiency plunges when requirements of individual users (e.g. consumer driver applications geared towards enterprise and consumer markets) decrease, OBS efficiency rises. On the other hand, if a Grid application demands lots of input and output, the OBS overhead rises in high gear compared to the one time overhead of setting up a circuit. One good strategy proposed is to use OBS for Grid applications with a low traffic throughput while using the OCS-strategy for Grid application which must interchange a huge amount of data between different Grid entities.

A major property of Grids is that different kinds of geographically distributed resources can be co-allocated (e.g. storage capacity, computation capacity, etc.) at the same time for a single application. This naturally has an impact on the network sustaining the Grid, which makes the scheduling, routing, dimensioning and corresponding network problems not as loosely-coupled as one would think.

Anycast routing

In the traditional network, it is common that the routing policy relies on shortest path routing. To convey data from one point to another, the shortest path in terms of the number of hops of the path is used. In a Grid scenario, this metric starts to lose its importance. The interest of the user of the Grid lies in the results of its job. Since different resources can process this job, the destination of the handling resource is of less significance. Anycast routing puts this idea to the test: users can submit their jobs with a specified service delivery without assigning a specified destination. This approach is especially useful for delivering consumer-oriented services over an optical network to a large number of users, as centralized job scheduling and Grid status monitoring on client devices can be avoided.

The use of anycast routing leads to a favorable combination of constraints for both the network parameters (bandwidth, delay) as well as for the resource parameters (e.g. processing availability and speed).

Advance reservations

The Grid environment requires that network information must be treated as resource information, so that we can know the network condition and transfer the data accordingly. Advanced reservation acts on this principle. It tries to guarantee simultaneous resource availability at the time of job processing. While different users will have different deadlines, a job scheduling strategy which satisfies the job deadlines is crucial. By reserving the necessary resources in advance, the scheduling entity of the Grid tries to reduce the blocking probability of the resources in the Grid and guarantees the availability of the resources. The interested reader is referred to [D. 5.4] for more information related to advance reservations.

4.2 Study of λ-switching in Lambda Grids

4.2.1 Suitability of the technique for particular applications/scenarios

The aim of this section is to establish the suitability of λ-switching compared to alternative switching techniques for particular scenarios through the analysis of a specific use-case. For this, we consider the circuit switching characteristics discussed in section 3.1 and also focus on the KoDaVis use-case presented in [D.3.1], where a group of spatially separated meteorologists are having a teleconferencing session over the KoDaVis visualization platform to view and analyze environmental data stored at a Grid site that is remote to all session participants. Due to the immense volume of environmental data, only an interactively selected (e.g. by the current speaker) portion of the data is retrieved and rendered at the visualization display of the meteorologist.

We first examine the applicability of circuit- resp. burst-switching to the QoS requirements posed by this use-case. Due to the interactivity feature, bounded delivery delay is a stringent requirement that could be equally satisfied by both OCS and OBS. Additionally, data visualization poses strict limits to delay variation (delay jitter). While delay variation in OCS is zero, in the case of OBS it is highly dependent on the load of the switching nodes (both edge and core). Even worse, the absence of a widely-accepted service differentiation scheme in OBS precludes guaranteeing a maximum delay jitter for the KoDaVis bursts. The same argumentation applies to the reliability requirements posed by interactivity: due to real-time selection, retrieval and rendering of visualization data, recovery of lost data is not an option; thus, minimization of data loss at the optical layer is the only practice towards guaranteeing seamless visualization quality. While advance reservation of computational resources and lightpaths guarantees negligible loss in the OCS case, the same is not true in the scenario of using OBS for the KoDaVis session.

Apart from the QoS argument that clearly indicates the suitability of a circuit-switched optical infrastructure solution to support Grid services with requirements similar to the KoDaVis use-case, it is worth investigating the relevant traffic pattern characteristics and whether these justifies the selection of a λ-switched or an OBS solution. In the use-case of focus, the session among the various participants is pre-planned and has a limited duration in the order of minutes or hours at maximum. Also, since liveness is a frequent feature of collaborative interactive applications, it is highly probable that continuous requests for remote data retrieval will lead to high utilization rates of the optical medium. Consequently, timely setting up and tearing down optical circuits for the session under study is possible, while high utilization of the reserved resources is achieved. This usage scenario fits well the intuition of using circuit-switching over burst/packet-switching at the optical layer.

4.3 Study of Optical Burst Switching in Lambda Grids

4.3.1 Suitability of the technique for particular applications/scenarios

Optical Burst Switching has a potential to bring several advantages for Grid-networking:

- Because different users have different traffic profiles, the bandwidth granularity offered by OBS allows efficient transmission of the user's task related data.
- Due to the separation of the of the control- and data plane, all-optical data transmission are allowed with particularly fast user- or application-initiated light-path setup.
- The electronic processing of the BHP at each node of the network infrastructure can incorporate Grid control layer properties. By checking the BHP, which can contain valuable Grid information, several user-controlled network functionalities can be supported.
    - QoS provisioning. By adopting the OBS control protocol the processing power of the router can be used to deploy advanced burst scheduling strategies, which are able to reduce delay while maintaining high bandwidth efficiency and low-burst loss rates.
    - Reliable multicasting. A reliable and scalable multicast protocol framework can be deployed in order to minimize the traffic load and to reduce recovery latency.
    - Resource discovery. When incoming data is discovered to be Grid data, the processing element of the router can immediately start searching for local resources.

4.3.2 Case Study: The Phosphorus WISDOM application

WISDOM (Wide In Silico Docking On Malaria) is an e-science application developed within the Phosphorus testbed. Typically, e-science applications require vast amounts of computation resources and vast amounts of storage resources, for the input/output data. The network infrastructure plays the role of transferring the input/output data to/from the computation resources (usually more than one), where the application is executed, from/to the storage sites. Thus, taking an abstraction of the problem, we have many-to-one and one-to-many (sometimes even many-to-many) transmissions of datasets of known size. In particular WISDOM application requires a many-to-one communication model, since the outputs of the experiments performed in more than one computation resources are aggregated at one data repository site at the end of the experiment (post-staging phase).

The routing of many-to-one and one-to-many connections (routing over trees) in WDM (OCS) networks is still an open issue in terms of efficiency and applicability. The so called "root" of such a tree may become a bottleneck, depending on the connectivity degree of the corresponding node. Assuming an underlined WDM network many wavelengths should be available in order to serve such a communication model. On the other hand, many-to-one and one-to-many connections are easily and efficiently handled in packet switched or burst switched networks (please refer to the Data Consolidation problem described in Section 2.2.2.2). OBS gives access to sub-wavelength bandwidth and enables the statistical multiplexing of flows over even a single wavelength. Various algorithms and signaling protocols proposed in the literature (Section 3.2.1 and Section 3.2.2) can be appropriate for such an application.

Another important characteristic of WISDOM application is that it does not require a network infrastructure that provides strict delay and delay jitter guarantees. Since the transferring of the output data from the computation to the storage resource is not the key part but the last part of the application, there are no strict time requirements. It goes without saying that a key requirement is that the data is not lost. Thus, the data cannot be transferred as best-effort traffic, but a mechanism that ensures the reception of the data at the destination should exist. Note that the output data can be broken in parts and transferred to the storage resource, the transmission rates can be at sub-wavelength speed, and more than one transmission (re-)trial can be performed. These characteristics seem appropriate for an OBS network. In particular OBS provides sub-wavelength granularity, and signaling mechanisms for retransmissions at the optical layer so as to avoid the loss of data in the core. Using an OCS-based architecture, instead of OBS, would result in inefficient use of the available bandwidth and would provide higher transmission standards than the ones required by the specific application.

4.3.3 Case Study: Optical Burst Switching in Phosphorus with Heterogeneous Traffic

4.3.3.1 *Problem Statement*

In [Chen07], an OBS based Grid, where Grid traffic co-exists with IP and/or 10GE traffic to achieve economy of scale was envisioned. The latency experienced by Grid jobs and IP/10GE traffic in the OBS networks was investigated. A decentralized Grid over OBS architecture was studied. The architecture includes a Grid aware burst assembler that separates traditional IP/10GE traffic and Grid jobs for processing. The Grid job and IP/10GE traffic are merged into the same burst assembly buffers. Simulation results have shown that Grid jobs consistently have lower latency than co-existing IP/10GE traffic, with a slightly elevated latency for IP/10GE traffic when the size of Grid jobs increases. In [Chen07], the inflexible unicast routing approach was implemented, where the destination is known before transmission.

As mentioned in section 4.1.5, the Grid user is interested in the job being processed regardless of the destination resource handling the job. In [DeLeenheer07b] several anycast routing algorithms were developed in the context of Grid over OBS to support the transmission of a single copy of a job when users submit their jobs without assigning a specified destination. Simulation results demonstrated that flexible destination assignment algorithm consistently outperforms the other two algorithms where the source assigns no explicit destination or the destination assigned by the source cannot be altered by any intermediate node. Also novel burst deflection techniques, incorporating both network and Grid state information, were introduced in [DeLeenheer07b]. The weighted Grid-resource availability (WGD), where the node examines all available Grid resources throughout the network to decide the forwarding port, appeared to be a good tradeoff between job blocking and average hop count.

In this section, we investigate a Grid over OBS architecture, where, as in [Chen07], Grid traffic co-exists with IP/10GE traffic, however, an anycast routing algorithm is implemented to route the Grid job bursts instead of the unicast routing. This implies that Grid jobs are separated from the IP/10GE traffic in the burst assembly buffers and each Grid job is transformed into a single burst. The soft assignment anycast routing algorithm and the weighted Grid-resource availability deflection policy proposed in [DeLeenheer07b] are implemented in this architecture. The performance of both Grid job and IP/10GE bursts is examined through simulation. Two performance metrics are presented, the job blocking probability and average job hop count. Also we examine the Influence of varying the computational resources load on the performance.

4.3.3.2 *Grid over OBS Architecture with Heterogeneous Traffic*

In this section, we describe a decentralized Grid over OBS architecture where Grid traffic co-exists with IP/10GE traffic. The architecture consists of Grid aware edge nodes and core nodes. Grid resources are connected to the Grid aware edge nodes. A Grid aware edge node performs a number of functionalities including, Grid Job Classification optical burst assembly and transmission, and Grid user to network as well as Grid resource to network signaling.

Figure 57 shows the details of a Burst Assembly unit in a Grid aware edge node. In the Grid Aware Burst Assembly Unit the Grid job traffic is separated from the IP/10GE traffic by the Traffic Classifier. The traditional IP/10GE traffic is forwarded to the Router /Classifier and Grid job traffic is forwarded to the Grid job Classifier

In the Router /Classifier, the destination edge router of the IP/10GE traffic packets is determined. Then they are routed to proper burst IP/10GE buffers. Hybrid burst assembly algorithm, which uses both time threshold T_{Th} and size threshold S_{Th} is used to assemble the collected IP/10GE traffic into a new burst. In this technique, a minimum burst size μ_{min} is introduced where all bursts must be equal to or larger than μ_{min}. Smaller bursts are padded to minimum burst size.

The Grid job classifier provides specialized Grid services based on user/application requirements. Grid jobs are separated from the IP/10GE traffic in the burst assembly buffers and each Grid job is transformed into a single burst as different routing algorithms is used for each traffic class.

While unicast defelection routing is used to route the IP/10GE bursts, an anycast routing algorithm is implemented to route the Grid job bursts. The no assignment anycast routing algorithm and the weighted Grid-resource availability deflection policy proposed in [DeLeenheer07b] are implemented in this architecture. Upon reception of a BHP, the core node checks if it has a sufficient free resources attached to deliver the job. Otherwise the node forwards the bursts according to a weighted function that directs a job towards the network region containing the nearest resources with the largest available capacity.

Distributed Grid resource discovery is implemented. As described in [DeLeenheer05], resources attached to edge nodes periodically send status packets, containing their free capacity, to the core nodes attached to them. These core nodes identify the resources as locally attached and exchange status packets between them to updating the value of free resource capacity attached to them. In case of contention, Grid job bursts are given priority over IP/10GE bursts. Each burst is allowed a maximum tolerable end-to-end delay, defined as the slack time, upon processing. If the slack time expires and the burst is not received yet, it will be dropped.

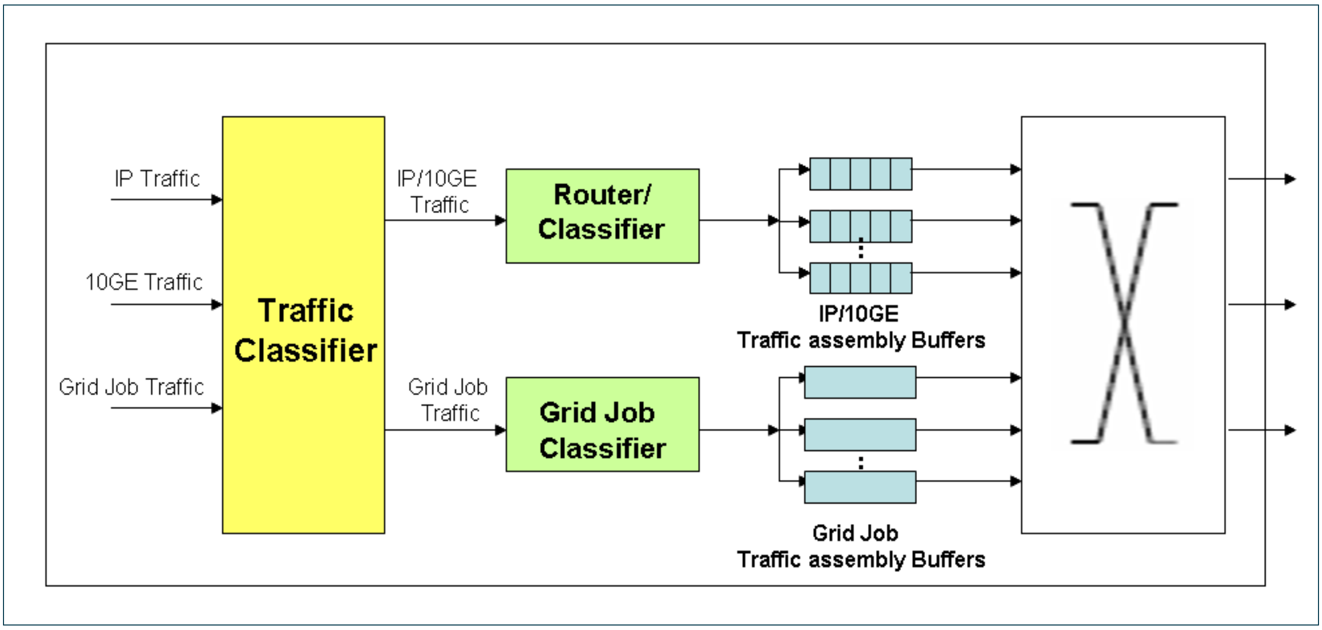


Figure 57: Grid Aware Burst Assembly Unit

4.3.3.3 *Performance Evaluation*

Simulation is carried out using the Phosphorus topology depicted in Figure 52. The topology contains 13 nodes and 19 bidirectional links. We assume all ports have 4 wavelengths each operating at 40 Gbps. We assume full wavelength conversion is available at each node, and a Just Enough Time (JET) reservation scheme. The best fit scheduling scheme is used in our architecture. No Fiber Delay Lines (FDL) are used.

Five nodes were randomly selected to function as computational Grid resource. The resources are assumed to have a processing capacity limited to 50 jobs in parallel. Both Poisson and self-similar job arrivals are considered. The self-similar traffic sources were simulated using aggregated ON–OFF Pareto-distributed sources. The Hurst parameter used for the aggregated traffic is H=0.9 which represents a high degree of burstiness. Grid job size is assumed to be exponentially distributed with a mean of 1 MB. Job execution time is also assumed to be exponentially distributed. Also both Poisson and self-similar interarrival time distributions are considered for the IP/10GE packets. Each burst is allowed 10 hops before its slack time expires.

Two performance metrics are presented, the burst dropping probability and burst hop count. Also we examine the Influence of Grid job size and computational resources load on the performance. Results are presented under two fixed IP/10GE traffic loads of 2 and 4 Eralngs, and varying Grid job traffic load. We assume a combined load equal to 80% of the computational resources capacity is generated.

Figure 58 compares the burst dropping probability experienced by the Grid job traffic and the IP/10GE traffic under varying Grid job traffic loads. It is clear that the Grid job traffic outperforms the IP/10GE traffic as a result of the priority given to the Grid jobs in case of contention. It can also be noticed that as the load increases the burst dropping probability increases. As the load increases, the probability of contention increases forcing bursts to travel further hops. However, traveling more hops, increases the probability that the burst's slack time might expires before the burst gets to its destination i.e. increasing the burst dropping probability.

Due to its burstiness, self-similar traffic, as shown Figure 58, increases the burst dropping probability compared to the Poisson (non-bursty) traffic.

Figure 59 compares the average hop count experienced by the Grid job traffic and the IP/10GE traffic. As mentioned above, as the load increases the probability that the burst's slack time might expire before the burst gets to its destination increases. This means that bursts with higher hop counts will be dropped and not considered in the average hop count. Therefore it is noticed form Figure 4 that the bursts average hop count decreases as the load increases. Also increasing the IP/10GE load form 2 to 4 decreases the average hop count. For a similar reason self-similar traffic results in decreasing the average hop count as load increases.

The performance under varying levels of generated resource loads is shown in Figure 60 and Figure 61. The Grid job traffic and the IP/10GE traffic loads are assumed to be constant at 2.5 for both. Generated resource loads up to 160% of the total resource capacity are examined. Grid job traffic suffers higher levels of burst dropping probability (Figure 60) as the generated resource load increases. Increasing the generated resource load results in increasing the number of hops the Grid job burst has to travel to get to a node with enough free resource capacity. As mentioned above traveling more hops increases the burst dropping probability. The IP/10GE bursts are also affected by increasing the generated resource load due to the contention created by Grid job bursts traveling more hops. Under resource loads higher than 120%, the Grid job bursts performance highly deteriorates resulting in a dropping probability higher than that suffered by the IP/10GE bursts. From Figure 61, it is noticed that the average hop count decreases as the generated resource load increases.

The performance under different Grid job sizes is shown in Figure 62 and Figure 63. The Grid job traffic and the IP/10GE traffic loads are also assumed to be constant at 2.5 for both. The generated resource load is also assumed to be 80%. The burst dropping probability (Figure 62) increases as the average job size increases. Scheduling larger bursts results in higher contention probability. This leads as discussed above to higher dropping probability and also a lower hop count as seen in Figure 63.

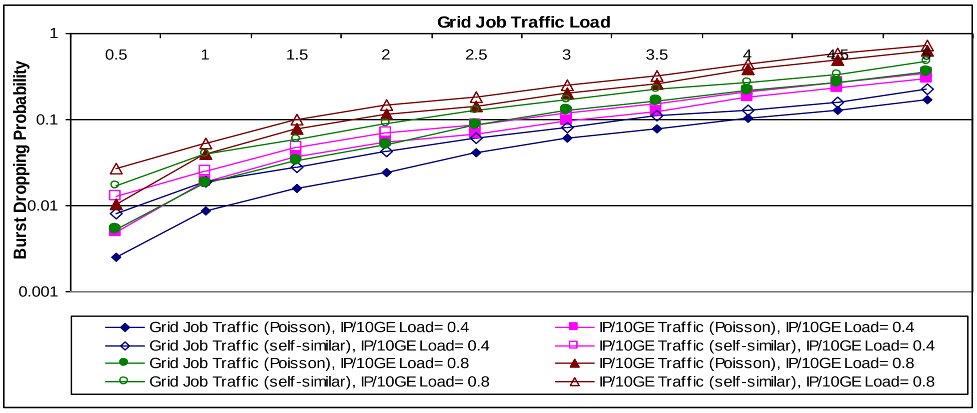


Figure 58: Average Burst Dropping Probability under varying Grid Job Traffic Load

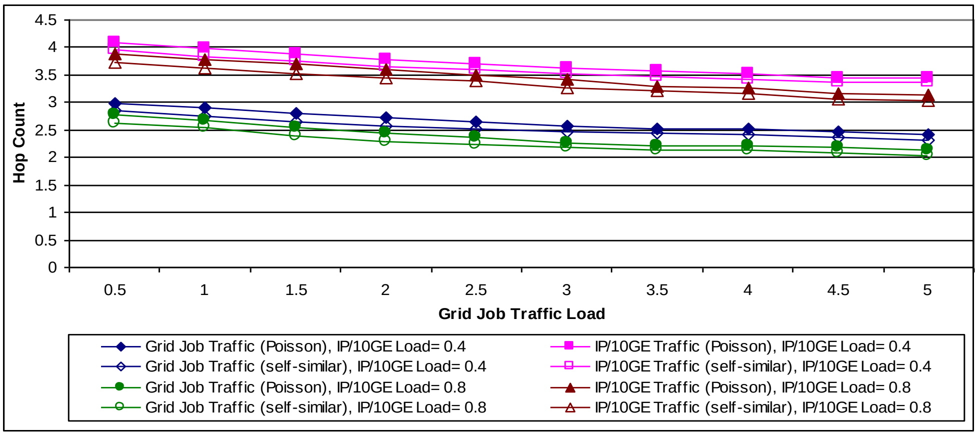


Figure 59: Average Burst Hop Count under varying Grid Job Traffic Load

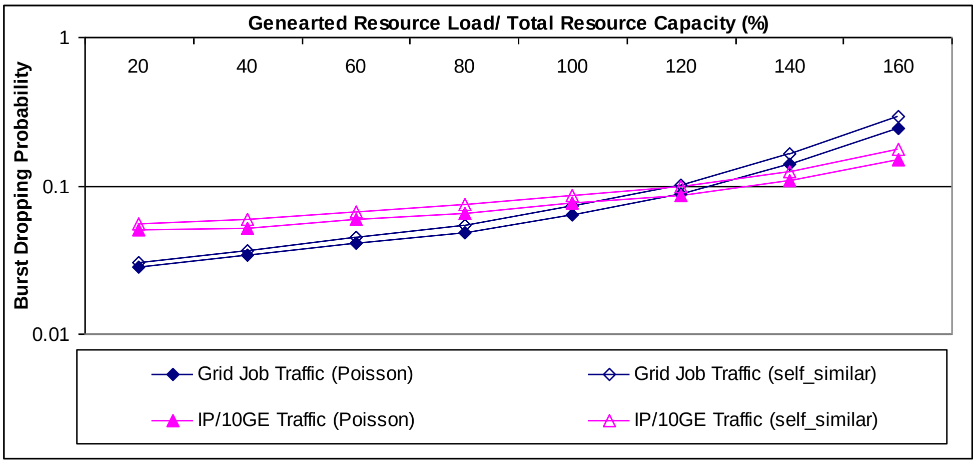


Figure 60: Average Burst Dropping Probability under Varying Generated Resource load

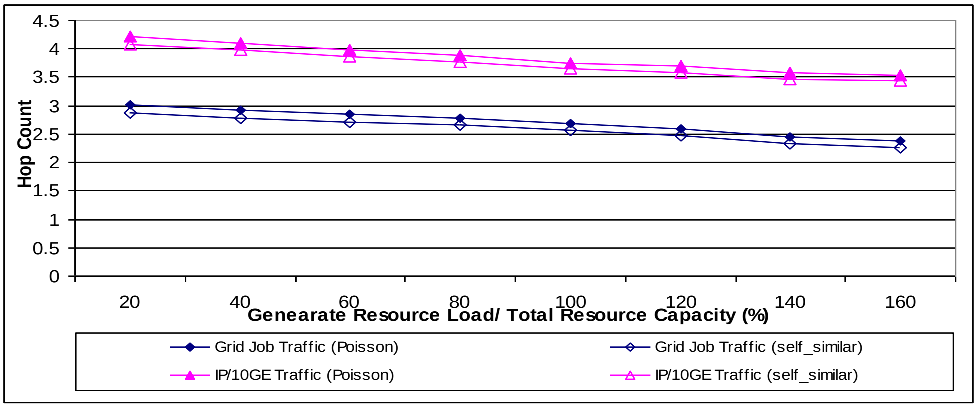


Figure 61: Average Burst Hop Count under Varying Generated Resource load

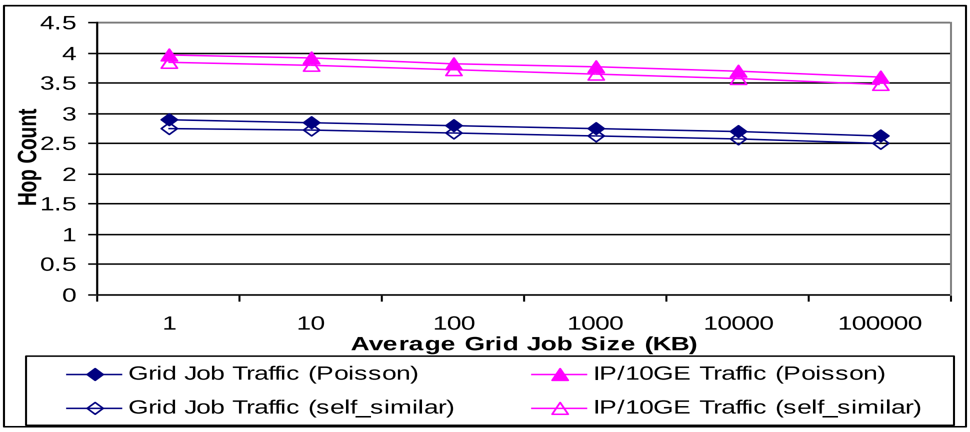


Figure 62: Average Burst Dropping Probability under Varying Average Gird Job Sizes

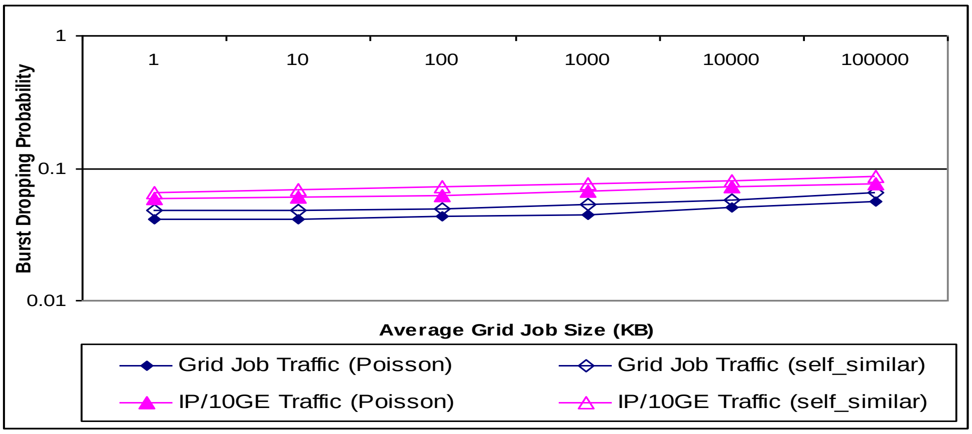


Figure 63: Average Burst Dropping Probability under Varying Average Gird Job Sizes

4.3.4 Case Study: Providing QoS for Anycasting over Optical Burst Switched Grid Networks

4.3.4.1 *Problem Statement*

The emerging Grid interactive applications require a user-controlled network infrastructure [Nejabati]. Control plane architectures for optical networks have been investigated by many researchers. A comprehensive review of the optical control plane for the Grid community can be found in [Jukan07]. Qualitity of Service (QoS) policies implemented in IP network do not work in the optical network, as the store-and-forward model does not exist [Kaheel02]. We thus see the need for an intelligent control plane in the optical network, which can provide the required QoS for Grid applications.

Delivering a Grid application effectively involves many parameters such as, design of efficient control plane architectures, algorithms for routing, providing QoS and resilience guarantees. Anycasting is similar to deflection routing, except for the fact that different destination can be selected instead of routing the burst to the same destination in another path. Routing can be accomplished by a label based control frame work [Kejie07] using optical core network, such as OBS. Anycasting allows the flexibility for the Grid job to effectively identify the destination that meets the QoS parameters. Incorporating an intelligent control plane and with the use of efficient signaling techniques, anycasting provides a viable communication paradigm to Grid applications.

In this section we present a mathematical framework to provide QoS for Grid Applications over optical networks. These QoS parameters include resource availability, reliability, propagation delay, and Quality of Transmission (QoT). These multiple services are needed to ensure the successful completion of a Grid job. With the help of link-state information available at each Network Element (NE), the bursts are scheduled to its next link. This de-centralized way of routing helps to provide optimal QoS and hence decreases the loss of Grid jobs due to multiple constraints.

4.3.4.2 *Service Parameters*

An anycast request can be denoted by $(s, D, 1)$ where s denotes the source, D the destination set and the last tuple indicates that a single destination has to be chosen from the set D. This notation is a generalization of manycast [Bathula08]. Let $m = |\mathrm{D}|,$ denote the cardinality of the set. Each Grid job has a service class and we hence define the service class set as $\mathrm{S} = \{\mathrm{S}_1, \mathrm{S}_2 \ldots \mathrm{S}_\mathrm{p}\}$ There is an associated threshold requirement, for which the QoS parameters should not exceed this condition. We define this threshold parameter as $\mathrm{T}(\mathrm{S_i})$, where $\mathrm{S_i} \in \mathrm{S}$

We define ω_j, η_j, γ_j and τ_j as the residual wavelengths, noise factor, reliability factor, and end-to-end propagation delay for Link j, respectively.In WR-OBS, the connection requests arrive at a very high speed while the average duration of each connection is only in the order of hundreds of milliseconds [Duser02]. To support such bursty nature of the traffic, it is always advisable to choose a path with more number of free wavelengths (least congested path). ω_j indicates the number of free (or residual) wavelengths available on link j. We consider an All-Optical Network (AON) architecture, where there is no wavelength conversion there, by resulting in Wavelength-Continuity Constraint (WCC). Let ω_i and ω_j be the two free wavelengths sets available on the links i and j respectively. Without loss of generality we assume that $W_i \cap W_j \neq \phi$ We propose to select a path towards the destination, with more number of free wavelengths. We use an operation $|\cap|$ which gives the common number of wavelengths on each link. If we assume that each uni-directional link can support 5 wavelengths, then $|W_i \cap \mathrm{W}_j|$ is an integer ≤ 5. The number of free wavelengths on the route is given by,

$$\omega_R = \left| \bigcap_{\forall \mathrm{i} \in \mathrm{R}} \mathrm{W_i} \right|, \tag{1}$$

where R denotes the route and ω_R represents the number of free wavelengths available. If $\omega_R = 0$, then the destination is said to be not reachable due to contention.

The noise factor is defined as ratio of input optical signal to noise ratio $(\mathrm{OSNR}_{i/p} \equiv \mathrm{OSNR}_\mathrm{i})$ and output optical signal to noise ratio $(\mathrm{OSNR}_{o/p} \equiv \mathrm{OSNR}_{\mathrm{i+1}})$ thus we have

$$\eta_\mathrm{j} = \frac{\mathrm{OSNR}_{i/p}}{\mathrm{OSNR}_{o/p}}, \tag{2}$$

where OSNR is defined as the ratio of the average signal power received at a node to the average ASE noise power at that node. The OSNR of the link and q-factor are related as,

$$q = \frac{2\sqrt{\frac{\mathrm{B_o}}{\mathrm{B_e}} OSNR}}{1+\sqrt{1+4OSNR}}\ , \qquad (3)$$

where B_o and B_e are optical and electrical bandwidths, respectively [Ramaswami04]. The bit-error rate is related to the q-factor as follows,

$$BER = 2erfc(\frac{q}{\sqrt{2}}). \qquad (4)$$

In our proposed routing algorithm, we choose a route that has minimum noise factor.

Thus the overall noise factor is given by,

$$\eta_R = \prod_{\forall i \in R} \eta_i , \qquad (5)$$

The other two parameters considered in our approach include, reliability factor and propagation delay of the burst along the link. The reliability factor of the link j is denoted by η_j . This value on the link indicates the percentage of the reliability of the link and its value lies in the interval [0, 1] The overall reliability of the route is calculated as the multiplicative constraint and is given by [Bathula08] [Jukan07],

$$\gamma_R = \prod_{\forall i \in R} \gamma_i , \qquad (6)$$

Propagation delay on the link j is denoted by _j and the overall propagation delay of the route R is given by,

$$\tau_R = \prod_{\forall i \in R} \tau_i , \qquad (7)$$

4.3.4.3 *Mathematical Framework*

In this section we provide the mathematical formulation for selecting the destination based on the above mentioned service parameters. We define Network Element Vector (NEV), that maintains information about the QoS parameters at each NE. This information is contained in the Optical Control Plane (OCP). In the distributed routing approach, current GMPLS routing protocols can be modified to implement the service information [Martinez07], [Farrel06]. A global Traffic Engineering Database (TED) at each OCP, which maintains an up-to-date picture of NEV.

Definition 1. We denote the network element vector for a link i as,

$$NEV_i = \left\langle \begin{matrix} \omega_i \\ \eta_i \\ \gamma_i \\ \tau_i \end{matrix} \right\rangle \tag{8}$$

Definition 2. Let NEV_i and NEV_j be the two network element information vectors of links i and j respectively, then we define a comparison 4 given by,

$$\left\langle \begin{matrix} \omega_i \\ \eta_i \\ \gamma_i \\ \tau_i \end{matrix} \right\rangle \preceq \left\langle \begin{matrix} \omega_j \\ \eta_j \\ \gamma_j \\ \tau_j \end{matrix} \right\rangle \tag{9}$$

The above equation implies that,

$$(\omega_i \geq \omega_j) \wedge (\eta_i \leq \eta_j) \wedge (\gamma_i \geq \gamma_j) \wedge (\tau_i \leq \tau_j) \tag{10}$$

Equation (10) is chosen such that, the path towards the destination has more number of residual wavelengths, low noise factor, high reliability and lower propagation delay.

Definition 3. The overall service information of a destination $d_n \in D$, $1 \leq n \leq m$ along the shortest path route $R(d_n)$ is given by,

$$NEV_{R(d_n)} = NEV_{R(d_n)}\left[s, h_1\right] \circ NEV_{R(d_n)}\left[h_1, h_2\right] \circ \ldots \circ NEV_{R(d_n)}\left[n_k, d_n\right] \tag{11}$$

$$NEV_{R(d_n)} = \left[\left| \bigcap_{\forall i \in R(d_n)} W_i \right|, \prod_{\forall i \in R(d_n)} \eta_i, \prod_{\forall i \in R(d_n)} \gamma_i, \sum_{\forall i \in R(d_n)} \tau_i \right]^T \tag{12}$$

where in (11) n_k represents the next hop node along the shortest-path. The operation $\circ$ performs $|\cap|$ on wavelengths sets, multiplication on noise factor, multiplication on reliability, and addition on propagation delay. Equation (12) represents the overall QoS information vector for the destination d_n

Definition 4. A destination d_n is said to be feasible for a given service requirement $\mathrm{T}^{(si)}$ if,

$$NEV_{R(d_n)} \preceq \mathrm{T}^{(si)} \tag{13}$$

The comparison of two multidimensional vectors using 4 follows from the notion of lattices [Przygienda95]. Using this ordering technique bursts can be scheduled to a destination that satisfies the service requirement if it is the best among the given set of destinations. In the next section we explain the proposed algorithm with the help of a network example.

4.3.4.4 *QoS Aware Anycasting Algorithm (Q3A)*

Below is the pseudo-code for the proposed algorithm. As we have considered service-differentiated scheduling, the threshold parameters of the particular service are know a-priori. In the initialization step, we consider the cardinality of the free wavelengths as the number of wavelengths the fiber can support. Other service parameters are considered to be 1 for multiplicative and 0 for additive, as indicated in the Line:1 of the algorithm.

For each destination $d_n \in D$, the next-hop node is calculated from the shortest-path routing (Line:2). By using the path algebra given in (11), the new network element information vector is computed and updated at the next hop node for d_n as n_k. A destination node d_n, is said to be qualified for the assigned Grid job, when $NEV_{R(d_n)}(s, n_k) \preceq T(S_i)$ (Line:4). If the required QoS are not met, then the anycast request is updated with the new destination set as given in Line:7. If the cardinality of D is zero, then the anycast request is said to be blocked for the given service threshold condition $T(S_i)$. However the same anycast request can satisfy another service S_j , $i \neq j$ with lower threshold requirements.

```
Input : $T(S_i), NEV_{R(d_n)}(s, n_{k-1})$
Output : $NEV_{R(d_n)}(s, n_k)$
  1: Initialization $NEV_{init} = [\omega_{max}, 1, 1, 0]^T$
  2: $Next\_Hop\_Node[s, d_n] = n_k$  /* $n_k$ is calculated from shortest path */
  3: $NEV_{R(d_n)}(s, n_k) \leftarrow NEV_{R(d_n)}(s, n_{k-1}) \circ NEV_{R(d_n)}(n_k, n_{k-1})$
  4: if $NEV_{R(d_n)}(s, n_k) \preceq T(S_i)$ then
  5: The path $(s, n_k)$ is feasible path and destination can be reached
  6: else
  7: Update the destination set $D \leftarrow D \setminus \{d_n\}$ /* Since route to $d_n$ does not satisfy the QoS
        requirement of the service $S_i$ */
  8: endif
  9: If $|D| = \phi$, then anycast request is blocked or lost
```

This algorithm calculates all the NEVs at intermediate and destination nodes. Intermediate NEVs check the threshold condition and discard the respective destination without further scheduling of the burst. Upon calculation of NEVs $(NEV_{R(d_n)} [,d_n])$ at all the updated destination set, these are re-ordered and the destination corresponding to the optimal NEV is selected. The equations below show the ordering technique used in selecting the final anycast destination.

$$NEV = \{NEV_{R(d_1)}, NEV_{R(d_2)}, \ldots, NEV_{R(d_p)}\} 1 \le \mathrm{p} \le \mathrm{n}, \text{(unsorted)} \qquad (14)$$

$$= \{NEV_{R(d'_1)}, NEV_{R(d'_2)}, \ldots, NEV_{R(d'_p)}\} \text{(sorted)} \qquad (15)$$

$$= \{NEV_{R(d'_1)} \preceq NEV_{R(d'_2)} \preceq \ldots \preceq NEV_{R(d'_p)} \preceq \mathrm{T}^{(\mathrm{Si})} \qquad (16)$$

From (16) d'_1 is the best destination among D that can meet the service requirement of S_i effectively.

This distributed Q3A approach can be implemented in a distributed way with help of a signaling approach [Jukan07]. BCP can be used to maintain the NEVs and update them as they traverse each NE. At each NE, TED is used to maintain the Traffic Engineering (TE) and can be modified to maintain the NEV.

4.3.4.5 *Network Example*

In this section we discuss the Q3A with help of a example to show the effectiveness of the algorithm in providing the QoS parameters. Consider the network shown in Figure 64. Consider the anycast request as (6, {2, 3, 4}, 1). The dotted lines in Figure 64 represent the shortest-path distance from source node 6 to the respective destination. The weights on each link represent, fiber distance in kms, noise factor, reliability factor and propagation delay in milli-seconds1. Table1 shows an set of free wavelengths on the links at the time of the anycast request.

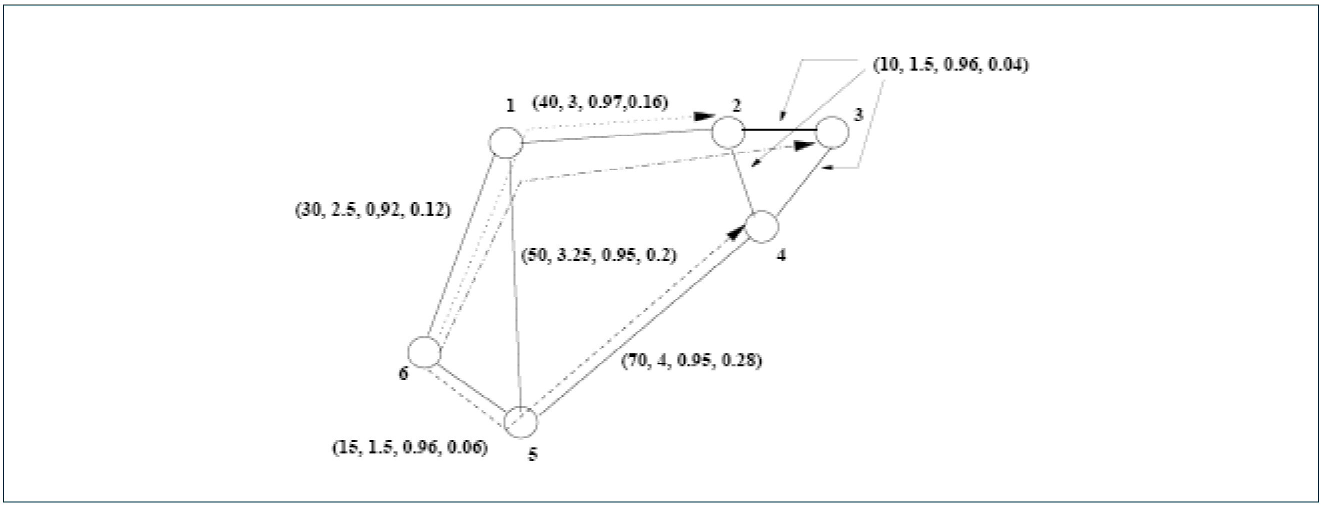


Figure 64: Network example used to explain the proposed Algorithm

#	**Link** (i→j)	**Residual Wavelength set** (W(i, j))
1	6 →5	{λ1, λ2, λ3, λ4, λ5}
2	5 →4	{λ1,λ2, λ3}
3	6→ 1	{λ1, λ2, λ5}
4	1 →2	{λ2, λ5}
5	2 →3	{λ3,λ4, λ5}

Table 7: Residual wavelengths available on links to all destinations

The NEVs for each destination can be calculated as given in below equations,

$$\begin{aligned} \mathrm{NEV}_{R(2)} &= [W(6,1),2.5,0.92,0.12]^T \circ [W(1,2),3,0.97,0.16]^T \\ &= [|W(6,2)|,7.5,0.89,0.28]^T \\ &= [1,7.5,0.89,0.28]^T \end{aligned} \qquad (17)$$

The free wavelengths on each link are obtained from Table 1 and the cardinality of the common wavelengths is represented in (17). This ensures the WCC in the all-optical networks, where there is an absence of wavelength converters. As the route towards the destination 3 shares the common path until node 2, NEV is given by,

$$\begin{aligned} \mathrm{NEV}_{R(3)} &= \mathrm{NEV}_{R(2)} \circ [W(2,3), 3, 0.97, 0.16]^T \\ &= [W(6,2), 7.5, 0.89, 0.28]^T \circ [W(2,3),1.5, 0.96, 0.04]^T \\ &= [1,11.5,0.85,0.32]^T \end{aligned} \qquad (18)$$

$$\mathrm{NEV}_{\mathrm{R}(4)} = \left[W(6,5), 1.5, 0.96, 0.04\right]^{T} \circ \left[W(5,4), 4, 0.95, 0.28\right]^{T}$$
$$= [3,6,0.91,0.32]^{T} \qquad (19)$$

From (17), (18), and (19) we observe, that destination 4 has an optimal QoS parameters (Except the propagation delay, which is slightly more than that of $\mathrm{NEV}_{\mathrm{R}(2)}$). This confirms the benefits of specifying the service requirements, whereby a destination can be chosen rather than selecting it at random.

4.3.4.6 *Simulation Results*

Simulation is carried out using the NSF network that consists of 14 nodes and 22 bi-directional links each at 10 Gbps. Each request consists of 3 destinations. Simulation results in Figure 65 compare the blocking probability for SPT (Shortest Path Tree) and Q3A. It is clear that Q3A outperformes that SPT.

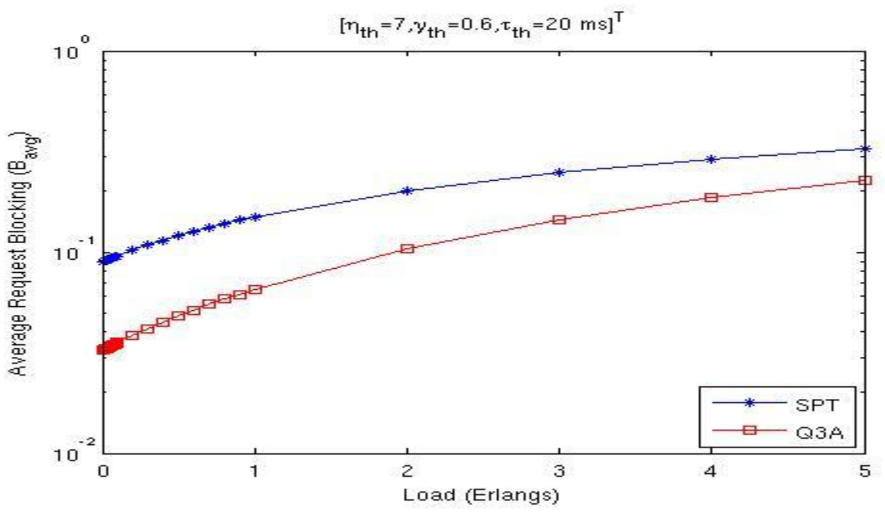


Figure 65: Avearge request Blocking of Q3A and SPT

In this section we discuss the provisioning of QoS for anycasting in Grid optical networks. By using the information vectors available at each NE, QoS parameters are computed. We have considered parameters that can be additive or multiplicative. Providing QoS to anycast communication allows the Grid application to choose a candidate destination according to its service requirements. This flexibility helps realize a user-controlled network. Our proposed algorithms also help in service-differentiated routing. Also Simulation results showes that the Q3A outperforms SPT.

4.4 Comparison of Techniques

The rapid development and deployment of fiber transmission technology and wavelength division multiplexing have made it probable that future networks will consist of some form of all-optical switching. Because of that development, transmission capacity got available in huge volumes, and it became clear that electronic processing at such line rates is very challenging. As a consequence, Optical Circuit Switching (OCS) was introduced. In OCS networks, bandwidth granularity is at the wavelength level since one or more wavelengths are allocated to a connection, while connectivity between source and destination is established using a two-way reservation. However, this form of transportation in optical networks can be very inefficient, especially in networks where applications generate bursty traffic. This is because OCS is neither sufficiently flexible nor bandwidth-efficient to support applications that require sub-wavelength bandwidth granularity in an on-demand fashion or for a short duration of time.

To tackle the problems associated with OCS, Optical Burst switching (OBS) has been introduced. OBS networks can be bufferless (unlike Optical Packets Switching (OPS) and can support users with different traffic profiles by electronically reserving the necessary bandwidth on a link only for the duration of a burst.

Although OBS is a very promising technique, it is not yet ready for deployment in large networks because of the high cost of OBS-supporting switches. Research is being conducted to hybrid techniques, which combine the merits and strengths of the basic switching technologies they are composed of. This work presents an attractive motivation to deploy a form of hybrid optical switching, namely Burst-over-Circuit-Switching (BoCS), as we demonstrate that it allows important cost-savings in comparison to the single technology switching solutions.

4.4.1 Case study

The lack of attention to the dynamics of OBS can lead to severe service degradation in terms of packet loss and congestion in some parts of the network. But on the other hand, by continuously keeping optical circuits alive for every connection between a source and a destination node, the network resources cannot reach their full potential. We will investigate how the introduction of circuits (and therefore change over to a hybrid optical switching technique) into the core of the network influences the network resources. We have implemented a form of the client-server hybrid optical technology: burst-over-circuit switching. This technique has OCS as the server- and OBS as the client technology. When computing a path from a source node to a destination node, the shortest path is calculated in the virtual topology presented to the client layer, where circuits are observed as one edge while they physically consist out of a consecution of links.

4.4.1.1 *Setup*

The network topology which we have considered consists of two binary-tree topologies of switches which are linked together by sharing the same root node. This type of network is representative for access/aggregation-type networks (e.g. PONs), while the center nodes are an abstracted form of the edge and/or core networks. The leaf nodes of the former tree create data, which has to be conveyed to the leaf nodes of the latter tree. The edge nodes use OBS as switching technology until a certain depth into the tree has been reached, where the switch aggregates the bursts and starts to send using OCS. In the remainder of this section we will address a topology with n leaf nodes and which has circuits starting from depth X in the tree as H_X^n. An example of a H_1^4 topology is presented in Figure 66.

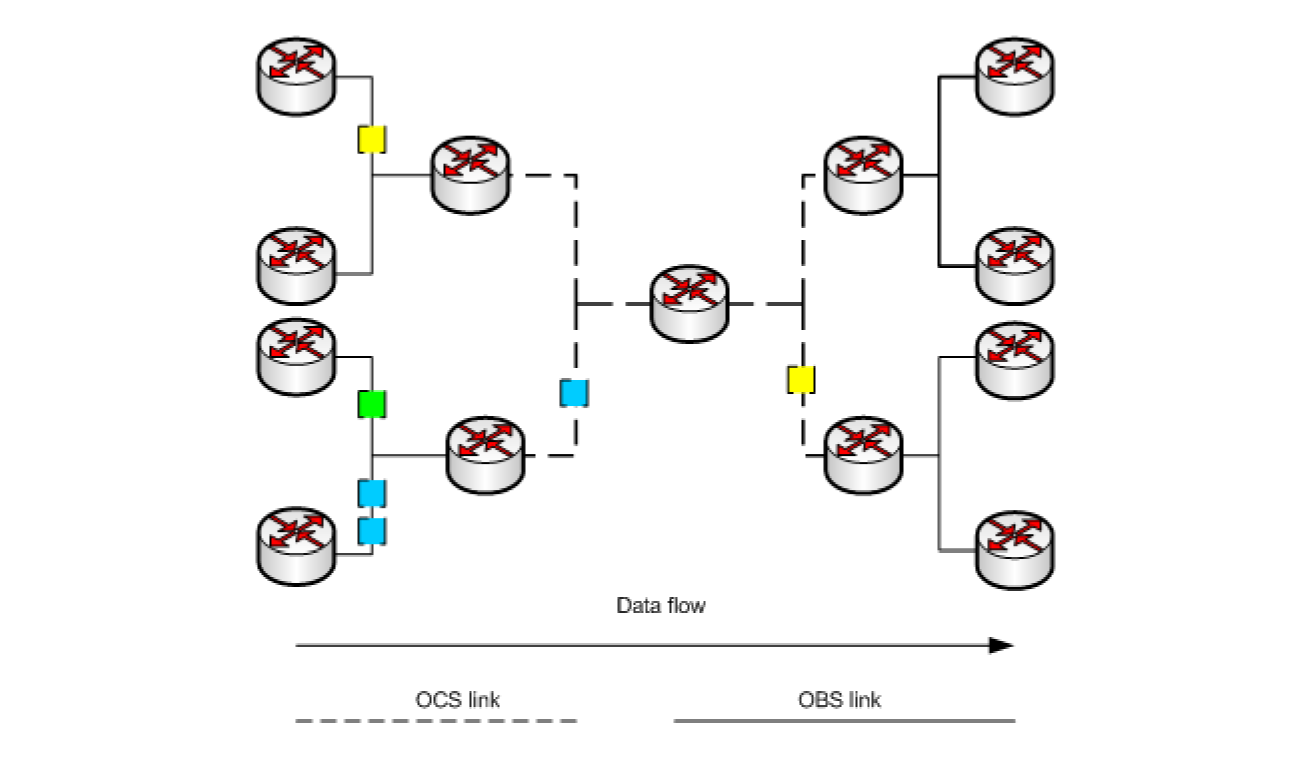


Figure 66: Example of a H_1^4 topology.

4.4.1.2 *Burst Loss*

In order to evaluate the hybrid switching technique, we have constructed a simulator. The simulation actually represents a Grid architecture supported by an optical network. Client nodes, which are in this case the leaf nodes of one sub tree, generate jobs which follow a Poisson process with an inter arrival rate λ. The data sizes of the jobs are also distributed as a Poisson process. These jobs are stuffed into an OBS packet and are sent to a resource node, which is situated at some leaf node of the opposite tree. These jobs are scheduled using a round-robin scheduler. Hence, we acquire a complete symmetrical case in every extent: every client has the same arrival rate, every link on a degree in the tree has the same amount of wavelengths and every destination (resource nodes) receives an equal part of the jobs.

The topology we have taken into consideration is a H_X^{16} network where x ranges from 0 (complete OBS network) to 4 (complete OCS network). What we expect is that the H_0^{16} topology will have the least amount of burst losses due to the statistical multiplexing property, in contrast to the H_4^{16} counterpart which should have a much larger burst loss that can be calculated using the well-known ErlangB formula. The H_X^{16} with

$x \in [1..3]$ burst loss rate should lie in between. This is exactly what our simulator demonstrates us and is shown in Figure 67 where the burst loss percentage is shown in function of an increasing load:

$$\rho = \frac{\lambda}{\mu\omega}.$$

In Figure 67 we can see that the difference between H_n^{16} burst loss and H_{n+1}^{16} burst loss rises with increasing n. This makes us conclude that the impact of OCS introduction into the core is not that large on burst loss when keeping the introduction insertion level x of H_X^{16} low.

Moreover we can make a mention that with a rising load the burst loss for every switching technique converges to a same fixed point. This leads us to the conclusion that a network should employ enough bandwidth, because the otherwise most expensive H_X^{16} shall perform almost even abominable as a cheaper H_X^{16}.

When the load is rather low$\rho \leq 0.6$), H_0^{16} , H_1^{16} and H_2^{16} are very close together which means they almost perform as well as a complete OBS network i.e. the H_0^{16} which is the lower bound for every other switching approach.

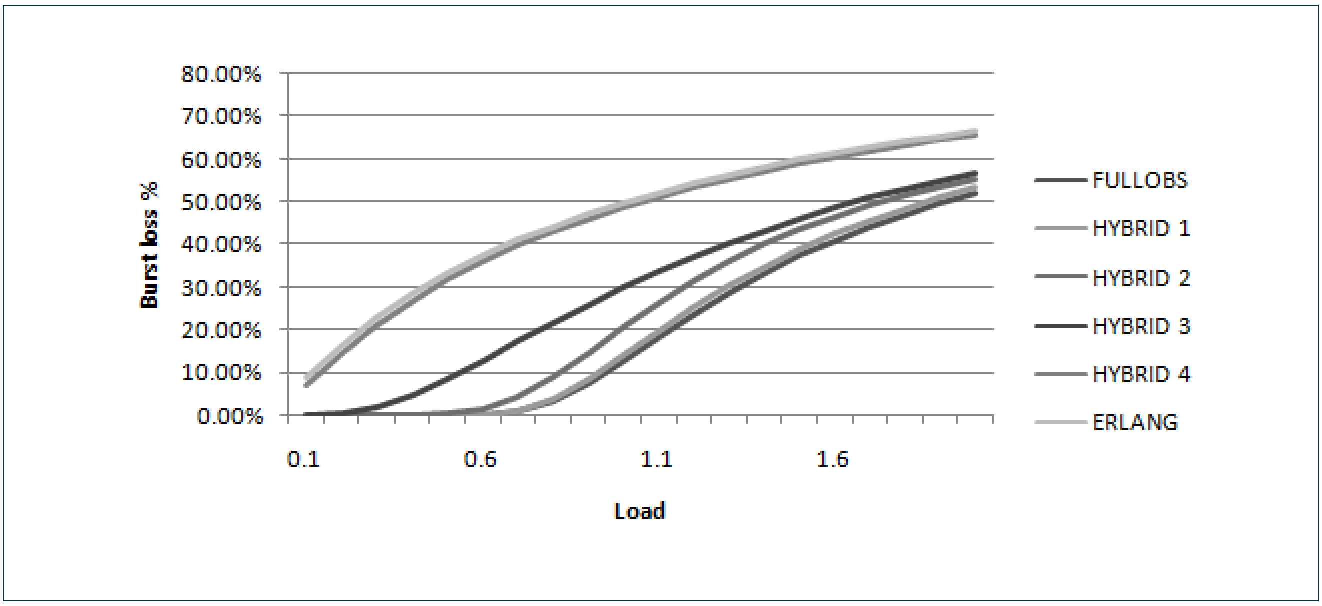


Figure 67: Simulation results for the different hybrid cases.

4.4.1.3 *Network capacity*

An important aspect of a switching technology is the allocation of network capacity to transfer data to and from end nodes, given a specified maximum loss rate. It is obvious to see that OBS will need fewer wavelengths than OCS due to the statistical multiplexing property. In what follows we will calculate how much wavelengths are needed per link for a certain H_X^{16} topology and show that we can optimize the network cost by using burst-over-circuit switching.

Bandwidth calculation for H_X^{16}

The probability that a burst is dropped in the network P_d is given by equations **Błąd! Nie można odnaleźć źródła odwołania.** and **Błąd! Nie można odnaleźć źródła odwołania.** where δ is the uniform burst drop probability per location where a burst drop event can take place and L is the length of the path from source to destination. The number of position on the path where a burst can be dropped is η

$$\eta = L$$

$$P_d = 1 - \prod_{1}^{\eta}(1-\delta)$$

This formula gives us a non-linear equation which we can solve for. Consequently, the individual link blocking probability δ can be modeled by the Erlang B formula **Błąd! Nie można odnaleźć źródła odwołania.**. This is based on the assumption that jobs are generated following a Poisson process as is also the case for the data size of these jobs. Now we can use numerical methods to solve the ErlangB formula equaling a target δ for its parameter λ as expressed in **Błąd! Nie można odnaleźć źródła odwołania.**. Also note that blocking of a switch is independent from all the other switches.

$$ErlangB(\lambda,\mu,\omega) = \frac{\frac{\left(\frac{\lambda}{\mu}\right)^{\omega}}{\omega!}}{\sum_{i=0}^{\omega}\frac{\left(\frac{\lambda}{\mu}\right)^{i}}{i!}}$$

$$ErlangB(\lambda,\mu,\omega) = \delta$$

From equation **Błąd! Nie można odnaleźć źródła odwołania.** we find λ_i .This is the rate parameter for the expected number of burst arrivals for the ith switch on the path from a client- to a resource node. When the number of wavelengths of the previous link is estimated, we can compute a new inter arrival timeλ_i by equation **Błąd! Nie można odnaleźć źródła odwołania.** and **Błąd! Nie można odnaleźć źródła odwołania.** as a result of the symmetry of the network.

$$ArrivalRate(\lambda,\mu,\omega) \;=\; 2(1 - ErlangB(\lambda,\mu,\omega)\lambda$$

$$\lambda_{\,i} = \;ArrivalRate(\lambda_i,\mu,\omega), \lambda_0 = \lambda_{clientNode}$$

This process is repeated, until we find all the wavelengths per link. This idea of estimating the net arrival rates based on loss rate estimates on previous links is known as a reduced load model.

Bandwidth Calculation for H_X^{16}

This computation differs from the previous because we have to incorporate the blocking events which are deserved by the circuits. When the circuit is inserted at level X, the number of possible places where a burst can be dropped becomes:

$$\eta = L - X + 1$$

With this η we can compute δ out of equation **Błąd! Nie można odnaleźć źródła odwołania.**. The number of wavelengths per OBS link follows from equation **Błąd! Nie można odnaleźć źródła odwołania.**. When the algorithm arrives at the beginning of the circuit, the number of wavelengths ω for that link follows out of equation **Błąd! Nie można odnaleźć źródła odwołania.** and **Błąd! Nie można odnaleźć źródła odwołania.**. This is because we have to calculate the number of wavelengths per destination.

$$ErlangB\left(\frac{\lambda}{2^X}, \mu, \widehat{\omega}\right) = \delta$$

$$\omega \ = \ 2^X \widehat{\omega}$$

4.4.1.4 *Discussion*

In Figure 68 and Figure 69 we see the number of wavelengths each switching technique requires relative to the total number of wavelengths which are needed in the $H_{\log_2} n^n$ case. We can see that the difference between the switching techniques plunges when the network gets larger. This conclusion is of value to the minimization of the overall network cost. The overall network cost K_n is given by **Błąd! Nie można odnaleźć źródła odwołania.**where K_c the cost per circuit wavelength is, K_o is the cost per OBS wavelength, N_c is the number of circuit wavelengths and finally N_o is the number of OBS wavelengths.

$$K_n \ = \ K_c N_c \ + \ K_o N_o$$

If we want that the hybrid switching is cost-effective than **Błąd! Nie można odnaleźć źródła odwołania.** should be satisfied.

$$K_c \ N_n \ \geq \ K_c \ N_c^h \ + \ K_c \ N_o^h$$

Hence, if we define α as $\frac{K_o}{K_c}$ then equation **Błąd! Nie można odnaleźć źródła odwołania.** should be satisfied for a hybrid approach to be cheaper than the pure OCS one. Note that OBS switching (at least with the current state-of-the art) is more expensive than OCS, so realistic values satisfy .

$$\alpha \leq \alpha_{max} = \frac{N_c - N_c^h}{N_o^h}$$

As an example, consider the results in Figure 70. We see that α increases with the level of circuit introduction. This lets us conclude that the relative cost of an OBS wavelength to an OCS wavelength can be larger with increasing circuit introduction. Or in other words the cost of an OBS wavelength can be at maximum αK_o for a specified network and switching technique, to have the same burst loss rate. In Figure 71 one can see that the development of a H_4^{64} topology the cost of an OBS wavelength is constrained by:

$$K_o \;\leq\; \alpha_{\max \square K_c}, \qquad \alpha_{max} = 3.9$$

For a known cost ratio α of OBS versus OCS switching, this study allows to find the BoCS variant which is cost-effective compared to pure OCS. The results show that hybrid approaches adopting an OBS/OCS combination have a larger margin than pure OBS: as long as the OBS technology does not mature and approaches OCS costs, a combination of OBS and OCS is advisable.

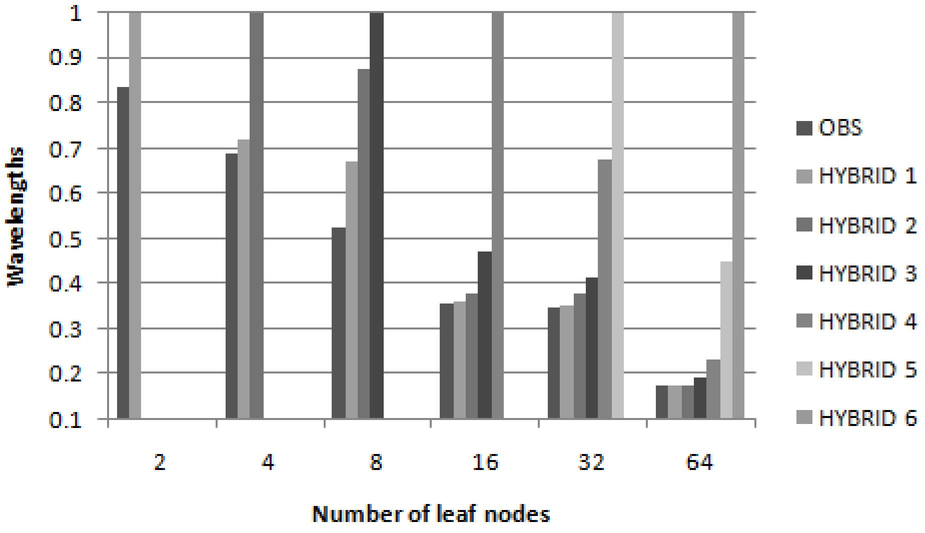


Figure 68: Relative wavelengths for the switching techniques to pure OCS, $\lambda = 10$

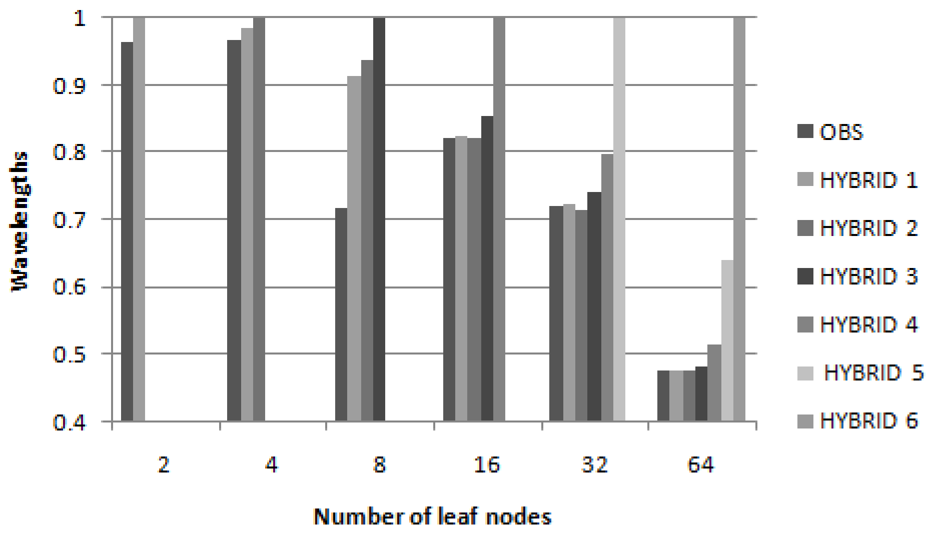


Figure 69: Relative wavelengths for the switching techniques to pure OCS, $\lambda = 100$

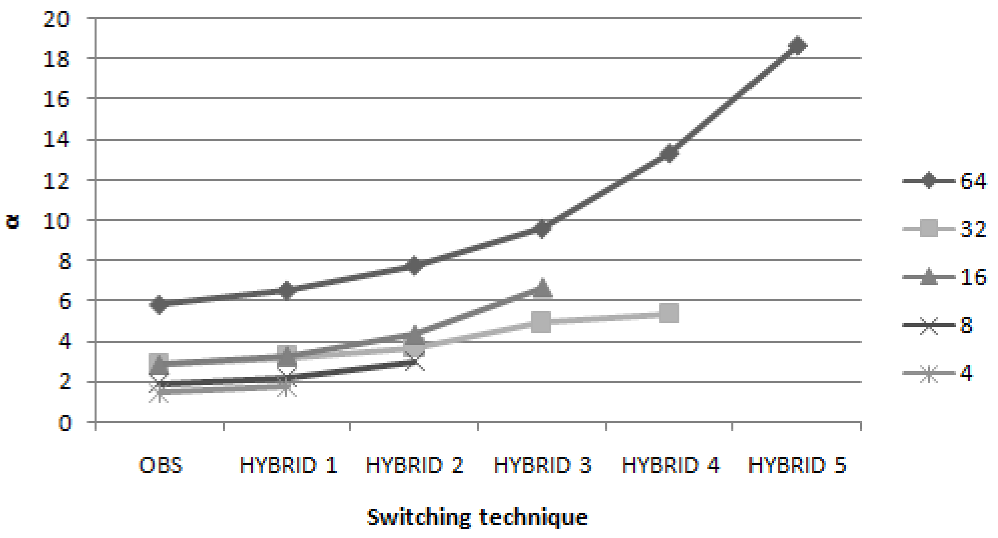


Figure 70: α_{max} computation for $\lambda = 10$

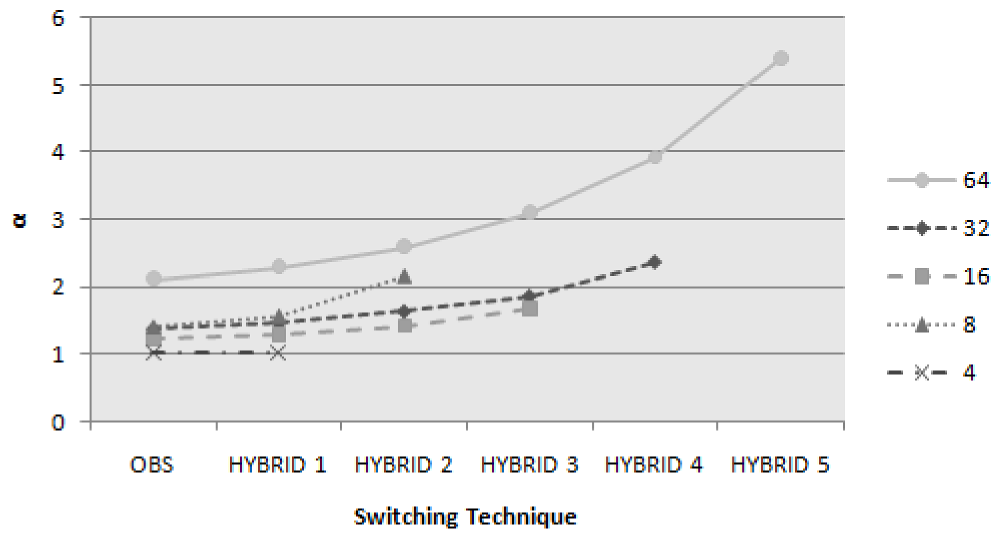


Figure 71: α_{max} computation for $\lambda = 100$

In general, hybrid optical network architectures, which combine two or more basic switching technologies, constitute a promising technique to optimize the overall network design. In this paper, a simulation analysis was used to evaluate one form of hybrid optical switching, namely burst-over-circuit switching. Results showed that depending on the expected load generated at the edge of a network, BoCS can be an attractive technology compared to the pure OBS and OCS alternative.

Finally, we have conducted an analytical network dimensioning study which concluded that for a specified type of network, the cost of a OBS wavelength to a OCS wavelength is bound by a parameter $\alpha_{\max}$ □ which in most cases is large enough to deploy BoCS in future networks. For the considered case study, we showed that as long as OBS switching costs are considerably more expensive than OCS, a hybrid BoCS approach is most cost effective.

5 References

[Alanyali01] M. Alanyali, “On Dynamic Wavelength Assignment in WDM Optical Networks”, pages 1–17. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[Ali02] M. Ali, L. Tancevski, "Impact of polarization-mode dispersion on the design of wavelength-routed networks", IEEE Photonics Technology Letters, Volume 14, Issue 5, pp. 720 - 722,May 2002.

[Ali04] R. J. Al-Ali, et al, “Analysis and provision of QoS for distributed grid applications,” Journal of Grid Computing, Vol.2, No. 2, pp. 163-182, June 2004.

[Baldine02] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, “JumpStart: A just-in-time signaling architecture for WDM burst-switched networks”, IEEE Communications, 40(2):82{89, February 2002.

[Barker07] R. M. Rahman, K. Barker, R. Barker, “A Predictive Technique for Replica Selection in Grid Environment”, Intl. Symposium on Cluster Computing and the Grid, pp 163-170, 2007.

[Bathula08] B. Bathula, ”QoS Aware Quorumcasting Over Optical Burst Switched Networks”, Ph. D. dissertation, Department of Electrical and Communication Engineering, Indian Institute of Science (IISc), Bangalore, India, 2008.

[BenYoo06] S.J. Ben Yoo, „Optical Packet and Burst Switching Technologies for the Future Photonic Internet”, Journal of Lightwave Technology, 24(12):4468-4492, Dec 2006.

[Birkan06] G. A. Birkan, “Practical Integrated Design Strategies for Opaque and All-Optical DWDM Networks: Optimization Models and Solution Procedures”, Telecommunication Systems, Vol. 31, No. 1, 2006.

[Biswas04] H. Shan, L. Oliker, W. Smith, R. Biswas, “Scheduling in Heterogeneous Grid Environments: The Effects of Data Migration”, Intl Conference on Advanced Computing and Communication, 2004.

[Bley04] A. Bley, T. Koch, R. Wessäly, “Large-scale hierarchical networks: How to compute an optimal architecture?”, Proc. 11th Int. Telecommun. Network Strategy and Planning Symposium (Networks 2004), Vienna, Austria, 13-16 Jun. 2004.

[Boost] Boost: http://www.boost.org/

[Buyya02] M. Maheswaran K. Krauter and R. Buyya, “A taxonomy and survey of grid resource management systems for distributed computing,” Software: Practice and Experience, Vol. 32, No. 2, pp. 135-164, February 2002.

[Cao02] X. Cao, J. Li, Y. Chen, C. Qiao, “Assembling TCP/IP packets in optical burst switched networks”, Proc. IEEE GLOBECOM, pp. 84-90, 2002.

[Carp03] T. Carpenter, D. Shallcross, J. Gannett, J. Jackel, A. Von Lehmen, “Maximizing the Transparency Advantage in Optical Networks”, in Proceedings of Optical Fiber Communication Conference, Vol. 2, Iss. 23-28, pp. 616-617, 2003.

[Chat07] B. Chatelain, S. Mannor, F. Gagnon and D.V. Plant, "Non-Cooperative Design of Translucent Networks", in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '07), pp. 2348-2352, Washington, DC, November 2007.

[Chen07] Y. Chen, W. Tang, P. Verma, “Latency in Grid over Optical Burst Switching with Heterogeneous Traffic”, Proc. HPCC, pp. 334-345, 2007.

[Choi98] H.A. Choi and E.J. Harder, “On Wavelength Assignment in WDM Optical Networks”, volume 468 of The Springer International Series in Engineering and Computer Science, pages 117–136. Springer US, 1998.

[Christo07] K. Christodoulopoulos, E. Varvarigos, C. Develder, M. De Leenheer, B. Dhoedt, “Job demand models for optical grid research”, Proc. 11th Int. IFIP TC6 Conf. on Optical Netw. Design and Modeling (ONDM2007), Athens, Greece; Lecture Notes in Computer Science, vol. 4534, pp. 127-136, May 2007.

[Christodou06] K. Christodoulopoulos, K. Vlachos, K. Yiannopoulos, E. A. Varvarigos, “Relaxing Delayed Reservations: An approach for Quality of Service differentiation in Optical Burst Switching networks”, IEEE International Conference on Broadband Communications, Networks and Systems, in proceedings of BROADNETS, vol.1, pp. 1-8, Oct. 2006, San Jose, CA, USA.

[Christodou07] K. Christodoulopoulos, E. Varvarigos and K. Vlachos, "A new Burst Assembly Scheme based on the Average Packet Delay and its Performance for TCP Traffic", Optical Switching and Networking, Elsevier, Vol. 4, No. 3-4, pp. 200-217, 2007 .

[Colle02] D. Colle, S. De Maesschalck, C. Develder, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, P. Demeester, "Data-centric optical networks and their survivability", IEEE J. Selected Areas in Commun., vol. 20, no. 1, pp. 6-20, Jan. 2002.

[D.2.1] Phosphorus Deliverable D.2.1, “The Grid-GMPLS Control Plane Architecture”.

[D.2.2] Phosphorus Deliverable D.2.2, “Routing and Signalling Extensions for the Grid-GMPLS Control Plane”.

[D.3.1] Phosphorus Deliverable D.3.1, “Use-cases, Requirements and Design of Changes and Extensions of the Applications and Middleware”.

[D.5.2] Phosphorus Deliverable D.5.2, “QoS-aware resource scheduling”.

[D.5.3] Phosphorus Deliverable D.5.3, “Grid job routing algorithms”.

[D.5.4] Phosphorus Deliverable D.5.4, “Support for advance reservations in scheduling and routing”.

[DeLeenheer05] M. De Leenheer et al, ``Anycast Routing in Optical Burst Switched Grid Networks'', 31st European Conference on Optical Communication (ECOC) 2005, Glasgow, Scotland, September 2005.

[DeLeenheer06] M. De Leenheer et al, “Anycast Algorithms Supporting Optical Burst Switched Grid Networks,” International Conference on Networking and Services (ICNS) 2006, July 16-19, 2006.

[DeLeenheer07] M. De Leenheer, C. Develder, F. De Turck, B. Dhoedt, P. Demeester, “Erlang reduced load model for optical burst switched grids”, Proc. 3rd Int. Conf. on Networking and Services (ICNS2007), Athens, Greece, 19-25 June 2007.

[DeLeenheer07b] Marc, D.L., et.la. ”Design and Control of Optical Grid Networks”, Proc. IEEE International Conference on BROADNETS, Raleigh, North Carolina, USA, September, pp. 107–115, 2007.

[DeLeenheer08] M. De Leenheer, C. Develder, J. Vermeir, J. Buysse, F. De Turck, B. Dhoedt, P. Demeester, „Performance Analysis of a Hybrid Optical Switch”, Proc. 12th Conference on Optical Network Design and Modelling (ONDM), Vilanova i la Geltru, Spain, Mar 2008.

[Develder08] C. Develder, B. Dhoedt, B. Mukherjee and P. Demeester, “On dimensioning optical grids and the impact of scheduling”, Photonic Network Commun., DOI: 10.1007/s11107-008-0160-z, 22 Aug. 2008

[Doulamis08] N. Doulamis, P. Kokkinos, E. A. Varvarigos, "Spectral Clustering Scheduling Techniques for Tasks with Strict QoS Requirements," Euro-Par 2008, pp. 478-488.

[Dueser02] M. Dueser and P. Bayvel. “Analysis of a dynamically wavelength-routed optical burst switched network architecture”, IEEE/OSA Journal of Lightwave Technology, 20:574-585, April 2002.

[Duser02] M. Duser, P. Bayel, ” Analysis of dynamically wavelength-routed optical burst switched network architecture”, J. Lightwave Technol., vol. 20, No. 4, pp. 574–585, 2002.

[EGEE] The Enabling Grids for E-sciencE project, http://www.eu-egee.org

[EGEE] F. Gagliardi, B. Jones B., F. Grey, M.E. Bégin and M. Heikkurinen, “Building an infrastructure for scientific Grid computing: status and goals of the EGEE project”, Philos. Trans. Series A, Math. Phys. Eng. Sci., vol. 363, no. 1833, pp. 1729–1742, 15 Aug. 2005.

[ENVI06] Enrico M. and M Vidondo, Experiences in Building a Pan-European Network over Dark-Fibre: Geant2, in Proceeding of TERENA Networking Conference, Catania, Italy, 2006.

[Farahmand04] F. Farahmand, Q Zhang and J.P. Jue, “A feedback-based Contention Avoidance Mechanism for Optical burst Switching Networks”, Workshop on Optical Burst Switching, 2004.

[Farrel06] A. Farrel, I. Bryskin,”GMPLS, Architecture and Applications”, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2006.

[Filho03] A. Filho and H. Waldman, “Strategies for Designing Translucent Wide-Area Networks”, in Proceedings of International Microwave and Optoelectronics Conference (IMOC), 931-936, 2003.

[Foster02] K. Ranganathan, I. Foster, “Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications”, Intl. High Performance Distributed Computing Sumposium (HPDC), pp. 352-358, 2002.

[Fulkerson62] L. Ford and D. Fulkerson, “Flows in Networks”, Princeton, NJ, 1962.

[G.8080] “ITU-T G.8080/Y.1304 Architecture for the Automatic Switched Optical Network (ASON)”, Internation Telecommunications Union, November 2001.

[Ge00] A. Ge, F. Callegati and L. Tamil, "On Optical Burst Switching and Self-Similar Traffic", IEEE Communications Letters, Vol. 4, No. 3. March 2000.

[Geant2] The GÉANT2 project, http://www.geant2.net

[GeoIP] Maxmind GeoIP, http://www.maxmind.com/app/locate_ip

[Gerstel00] O. Gerstel, R. Ramaswami, G. H. Sasaki, “Cost-effective traffic grooming in WDM rings”, IEEE/ACM Trans. Networking, vol. 8, no. 10, pp. 618–630, Oct. 2000.

[Giles00] G. Flake, S. Lawrence, and C. L. Giles. “Efficient Identification of Web Communities.” In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000), pp. 150—160. New York: ACM Press, 2000.

[GOBS] Grid Optical Burst Switched Networks: http://www.ogf.org/Public_Comment_Docs/Documents/Jan-2007/OGF_GHPN_GOBS_final.pdf

[Holler06] H. Höller, S. Voß, “A heuristic approach for combined equipment-planning and routing in multi-layer SDH/WDM networks”, European J. of Operational Research, vol. 171, no. 3, pp. 787-796, Jun. 2006.

[Jukan07] A. Jukan, ” Optical Control Plane for the Grid Community”, J. IEEE Communications Surveys & Tutorials, vol.9, No. 3, pp. 30–44 , 2007.

[Jukan07b] A. Jukan, G. Franzl, ”Path selection methods with multiple constraints in service guaranteed WDM networks”, J. IEEE/ACM Trans. Networking, vol. 12, No. 1, pp. 59–72, 2004.

[Kaheel02] A. Kaheel, T. Khattab, A. Mohamed, H. Alnuweiri, ”Quality-of-service mechanisms in IP-over-WDM networks” J. IEEE Communications Magazine, vol. 40, No.12, pp. 38–43, 2002.

[Kantarci05] Burak Kantarci, Sema F. Oktug, Tülin Atmaca, “Analyzing the Effects of Burst Assembly in Optical Burst Switching under Self-Similar Traffic”, AICT/SAPIR/ELETE, pp. 109-114, 2005.

[Kejie07] Kejie Lu., et.la.” An anycast routing scheme for supporting emerging grid computing applications in OBS networks”, Proc. IEEE International Conference on Communication (ICC), Glasgow, UK, pp.2307–2312, June 2007.

[Kokkinos07] P. Kokkinos, E. Varvarigos, N. Doulamis, “A Framework for Providing Hard Delay Guarantees in Grid Computing”, eScience, pp. 271-278, 2007.

[Kokkinos08] P. Kokkinos, K. Christodoulopoulos, A. Kretsis, E. A. Varvarigos, "Data Consolidation: A Task Scheduling and Data Migration Technique for Grid Networks" CCGRID, 2008, pp. 722-727.

[Leahy93] Z. Wu and R. Leahy, “An Optimal Graph Theoretic Approach to Data Clustering: Theory and its Application to Image Seg-mentation,” IEEE Trans. Pattern Analysis and Machine Intelli-gence, Vol . 15, No. 11 pp. 1101-1113, Nov. 1993.

[Liu03] J. Liu, N. Ansari, T. Ott, "FRR for Latency reduction and QoS Provisioning in OBS Networks", IEEE JSAC, Vol. 21, No. 7, Sept. 2003.

[MacQueen67] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations”, Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability, Berkeley, University of California Press, vol. 1, pp. 281-297, 1967.

[Maesschalck03] S. De Maesschalck, et al., “Pan-European optical transport networks: an availability-based comparison”, Photonic Network Commun., vol. 5, pp. 203-225, May 2003.

[Mannie03] E. Mannie, “Generalized Multi-Protocol Label Switching Architecture”, RFC 3945, Internet Engineering Task Force, October 2004

[Martinez07] R. Martinez, F. Cugini, N. Andriolli, L. Wosinska, J. Comellas, “Challenges and Requirements for Introducing Impairment-Awareness into Management and Control Planes of ASON/GMPLS WDM Networks”, IEEE Communication Magazine, vol. 44, No. 12, pp. 76–85, 2007.

[MMS07] E.Q. Vieira Martins, M.B. Pascoal and J.L.E. Santos, “A New Algorithm for Ranking Loopless Paths”, Research Report, CISUC, May 1997.

[Morato01] D. Morato, J. Aracil, L. Diez, M. Izal, E. Magana, "On linear prediction of Internet traffic for packet and burst switching networks", In Proceedings of 10th International Conference on Computer Communications Networks, pp. 138-143, 2001.

[Morea04] A. Morea, H. Nakajima, L. Chacon, Y. Le Louedec, J.-P. Sebille, “Impact of the reach distance of WDM systems on the cost of translucent optical networks”, in Proceedings of Telecommunications Network Strategy and Planning Symposium, pp. 321-325, June 2004.

[Mukherjee96] B. Mukherjee, D. Banerjee, S. Ramamurthy, A. Mukherjee, “Some principles for designing a wide-area WDM-network”, IEEE/ACM Trans. on Networking, vol. 4, no. 5, pp. 684–696, Oct. 1996.

[Nejabati] R. Nejabati, ”Grid Optical Burst Switched Networks (GOBS)”, www.ogf.org.

[Nemhauser99] G.L. Nemhauser and L.A. Wolsey, “Integer and Combinatorial Optimization”, Wiley-Interscience, Nov 1999.

[ns2] Optical burst switching simulator based on ns-2: http://wine.icu.ac.kr/~obsns/index.php

[OFC02] F. Xue, S. Yao, B. Mukherjee and S.J.B Yoo, “The performance improvement in optical packet switched networks by traffic shaping of self-similar traffic”, Optical Fiber Communication (OFC) Conf., 2002, pp. 218-219.

[Ostring01] S. Ostring and H. Sirisena, "The Influence of Long-range Dependence on Traffic Prediction", In Proceedings of ICC, 2001.

[Papadimitr03] G.I. Papadimitriou, C. Papazoglou, A.S. Pomportsis, “Optical Switching: Switch Fabrics, Techniques, and Architectures”, Journal of Lightwave Technology, 21(2):384–405, Feb 2003.

[Pickavet99] M. Pickavet, P. Demeester, “Long-term planning of WDM networks: a comparison between single period and multi-period techniques”, Photonic Network Commun., vol. 1, pp. 331–346, Dec. 1999.

[Przygienda95] A. Przygienda,”Link state routing with QoS in ATM LANs”, Ph. D. dissertation, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1995.

[Qiao99] C. Qiao and M. Yoo, “Optical burst switching (OBS)–a new paradigm for an optical Internet,” J. of High Speed Networks, vol. 8, no. 1, pp. 69–84, 1999.

[RamaFe99] B. Ramamurthy, H. Feng, D. Datta, J.P. Heritage, B. Mukherjee, “Transparent vs. opaque vs. translucent wavelength-routed optical networks”, in Proceedings of Optical Fiber Communication Conference, pp. 59-61, 1999.

[Ramaswami04] R. Ramaswami, N. Kumar, ”Optical Networks”, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2004.

[Rosberg03] Z. Rosberg, H.L. Vu, M. Zukerman, J. White, “Blocking probabilities of optical burst switching networks based on reduced load fixed point approximations”, Proc. 22nd Annual Joint Conf. of the IEEE Computer and Commun. Societies (Infocom 2003), San Francisco, CA, USA, Vol. 3, pp. 2008-2018, 30 Mar. - 3 Apr. 2003.

[Rouskas07] C. Castillo, G. Rouskas and K. Harfoush “On the Design of Online Scheduling Algorithms for Advance Reservations and QoS in Grids,” IEEE International Conference on Parallel and Distributed Processing, pp. 1-10, March 2007..

[Savasi07] M. Savasini, P. Monti, M. Tacca, A. Fumagalli and H. Waldman, “Regenerator Placement with Guaranteed Connectivity in Optical Networks”, in Proceedings of 11th International IFIP Conference on Optical Network Design and Modeling, pp. 438-447, May 2007.

[Seklou08] K. Seklou, E. Varvarigos, “Fast Reservation Protocols for Latency Reduction in Optical Burst-Switched Networks Based on Predictions”, International Conference on Networking (ICN) 2008.

[Seklou08b] K. Seklou, A. Sideri, E. Varvarigos, “New Assembly Techniques and Fast Reservation Protocols for Optical Burst Switched Networks Based on Traffic Prediction”, submitted to Computer Communications journal.

[Shen02] G. Shen, W. Grover, T. Cheng, and S. Bose, "Sparse placement of electronic switching nodes for low blocking in translucent optical networks", Journal of Optical Networking, Vol. 1, Iss. 12, pp. 424-441, December 2002.

[Shiraz01] H. Ghafouri-Shiraz, G. Zhu, and Y. Fei, “Effective Wavelength Assignment Algorithms for Optimizing Design Costs in SONET/WDM Rings”, Journal of Lightwave Technology, 19(10):1427–1439, Oct 2001.

[Sideri07] A. Sideri, E. Varvarigos, “New Assembly Techniques for Optical Burst Switched Networks Based on Traffic Prediction”, Optical Network Design and Modeling (ONDM), 2007.

[Simmons06] J.M. Simmons, “Network Design in Realistic “All-Optical” Backbone Network”, IEEE Communications Magazine, Vol. 44, Issue 11, pp. 88-94, November 2006.

[SMTFW07] Savasini, M.S., et al., Regenerator Placement with Guaranteed Connectivity in Optical Networks, in Optical Network Design and Modeling, p. 438-447, 2007.

[Stock01] R. Duda, P. Hart, D. Stock, “Pattern Classification”, John Willey and Sons, 2001.

[Subraman96] S. Subramaniam, M. Azizoglu, A.K. Somani, “All-Optical Networks with Sparse Wavelength Conversion”, IEEE/ACM Transactions on Networking, 4(4):544–557, Aug 1996.

[Teng05] J.Teng and G.Rouskas, “Routing Path optimization in optical burst switched networks”, ONDM 2005, p 1-10.

[Thodime03] G.P.V. Thodime, V.M. Vokkarane and J.P. Jue, “Dynamic Congestion-Based Load Balancing Routing in Optical Burst-Switched Networks”, IEEE GLOBECOM, Dec. 2003, pp. 2628-2632.

[Thysebaert05] P. Thysebaert, F. De Turck, B. Dhoedt, P. Demeester, “Using divisible load theory to dimension optical transport networks for Grid excess load handling”, Proc. Int. Conf. on Autonomic and Autonomous Systems & Int. Conf. on Networking and Systems (ICAS/ICNS 2005), Papeete, Tahiti, 23-28 Oct. 2005.

[Thysebaert07] P. Thysebaert., et.al."Scalable dimensioning of Resilient lambda Grids" Future Generation Computer Systems, Volume 24 Issue 6, 2007

[Treicher97] J. Treicher, C. Johnson Jr. and M. Larimore, “Theory and Design of Adaptive Filters”, New York: Wiley 1987. The Network Simulator – ns-2: http://www.isi.edu/nsnam/ns/index.html

[Turner99] J. Turner, “Terabit burst switching,” J. High Speed Networks, vol. 8, pp. 3–16, 1999.

[VanBreuse06] 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 J. Selected Areas in Commun., vol. 24, no. 4, supplement, pp. 13-26, Apr. 2006.

[VanPar01] W. Van Parys, P. Arijs, O. Antonis,P. Demeester, "Quantifying the benefits of selective wavelength regeneration in ultra long-haul WDM networks", in Proceedings of Optical Fiber Communication Conference and Exhibit (OFC '01), 2001.

[Varvarigos05] E. Varvarigos, N. Doulamis, A. Doulamis, and T. Varvarigou “Timed/Advance Reservation Schemes and Scheduling Algorithms for QoS Resource Management in Grids,” Engineering the Grid: Status and Perspective, Edited by B. Di Martino, J. Dongarra, A. Hoisie, L. T. Yang, and H. Zima Chapter 22, pp. 355-377, American Scientific Publishers, 2005.

[Varvarigos08] E.A. Varvarigos, V. Sourlas, K. Christodoulopoulos, “Routing and Scheduling Connections in Networks that Support Advance Reservations” accepted for publication in Elsevier Computer Networks.

[Varvarigos97] E.A. Varvarigos, V. Sharma, “The ready-to-go virtual circuit protocol: a loss-free protocol for multigigabit networks using FIFO buffers”, IEEE/ACM Transactions on Networking, vol.5, (no.5): pp.705-18, Oct. 1997.

[Varvarigos98] E.A. Varvarigos and V. Sharma, “An efficient reservation connection control protocol for gigabit networks”, Computer Networks and ISDN Systems 30, pp. 1135–1156, 1998.

[Veselic03] I. Nakic and K. Veselic, “Wielandt and Ky-Fan Theorem for Matrix Pairs,” Linear Algebra and its Applications, Vol. 369, No. 17, pp.77-73, August 2003.

[Vokkarane03] V. Vokkarane, K. Haridoss, J. Jue, "Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks", Proc. Opticomm, pp. 125-136, 2003.

[Vokkarane03b] V. M. Vokkarane, J. P. Jue, “Prioritized burst segmentation and composite burst-assembly techniques for QoS support in optical burst-switched networks,” IEEE Journal on Selected Areas in Communications, vol. 21, issue 7, pp.1198-1209, September 2003.

[Wei00] J. Y. Wei and R. I. McFarland, “Just-in-time signaling for WDM optical burst switching networks”, Journal of Lightwave Technology, 18(12):2019{2037, December 2000.

[Wen05] H. Wen, H. Song L. Li and S Wang “Load Balancing contention resolution in OBS networks based on GMPLS”, Int. J High Performance Computing and Networking 2005, Vol. 3 No 1 pp 25-32.

[Xu03] J. Xu, C. Qiao, J. Li, and G. Xu, “Efficient channel scheduling algorithms in optical burst switched networks,” in Proc. INFOCOMM 2003, vol. 3, pp. 2268–2278.

[Yang05] X. Yang, B. Ramamurthy, “Sparse Regeneration in Translucent Wavelength-Routed Optical Networks: Architecture, Network Design and Wavelength Routing”, Springer Journal of Photonic Network Communications, 2005.

[Yang05b] L Yang and G. Rouskas, “Adaptive path selection in optical burst switched networks”, IEEE/OSA Journal of Lightwave Technology Vol 24 NO 8, Aug. 2006.

[YEK03]	Yetginer, E. and E. Karasan, Regenerator Placement and Traffic Engineering with Restoration in GMPLS Networks, Photonic Network Communications, Vol. 6**,** Iss. 2, 2003.
[Yetgin03]	E. Yetginer, and E. Karasan, “Regenerator Placement and traffic Engineering with Restoration in GMPLS Networks”, Photonic Network Communications, Vol. 62, pp.139-149, 2003.
[Yu02]	X. Yu, Y. Chen, and C. Qiao, "Study of traffic statistics of assembled burst traffic in optical burst switched networks," In Proceeding of Opticomm, pp. 149-159, 2002.
[Zang02]	H. Zang, R. Huang, and J. Pan, “Designing a Hybrid Shared-Mesh Protected WDM Networks with Sparse Wavelength Conversion and Regeneration”, in Proceedings of SPIE Asia-Pacific Optical Communications Conference (APOC), October 2002.
[Zapata03]	A. Zapata and P. Bayvel, “Dynamic wavelength-routed optical burst-switched networks: scalability analysis and comparison with static wavelength-routed optical networks”, Optical Fiber Communication (OFC) Conf., 2003, pp 212-213.
[Zervas08]	G. Zervas, M. De Leenheer, L. Sadeghioon, D. Klonidis, R. Nejabati, D. Simeonidou, C. Develder, B. Dhoedt, P. Demeester, “Multi-Granular Optical Cross-Connect: Design, Analysis and Demonstration”, Submitted to IEEE Journal on Selected Areas in Communications.
[Zhang06]	Y. Zhang et. al. “Scalable Grid Application Scheduling via Decoupled Resource Selection and Scheduling,” IEEE 6th Intl Conference on Cluster Computing and the Grid (CCGRID), pp. 568-575, May 2006.
[Zhu03]	K. Zhu, H. Zang, B. Mukherjee, “A comprehensive study on next-generation optical grooming switches”, IEEE J. Selected Areas in Commun., vol. 21, no. 7, pp. 1173-1186, Sep. 2003.
[Zini02]	W. Bell, D. Cameron, L. Capozza, A. Millar, K. Stockinger, F. Zini, “Simulation of Dynamic Grid Replication Strategies”, OptorSim, LNCS, Vol. 2536 , pp. 46-57, 2002.

6 Acronyms

[3R]	Retiming, Reshaping and Reamplifying
[AON]	All-Optical network
[ASE]	Amplified Spontaneous Emission Noise
[BCH]	Burst Control Header
[BCP]	Burst Control Packet
[BER]	Bit Error Rate
[BHP]	Burst Header Packet
[BoCS]	Burst over Circuit Switching
[CD]	Chromatic Dispersion
[CPU]	Central Processing Unit
[CST]	Circuit Setup Signaling Time
[DBR]	Average Packet Burstification Delay to Burst Size ratio
[DC]	Data Consolidation
[D-ND]	Deployed Network Dimensioning
[EGEE]	Enabling Grids for EsciencE
[FDL]	Fiber Delay Lines
[FLOP]	Floating Point Operations
[FR]	Fast Reservation
[FWM]	Four-Wave Mixing
[G^{2}MPLS]	Grid-enabled GMPLS
[G-ND]	Greenfield Network Dimensioning
[IA-G-ND]	Impairment-Aware Greenfield Network Dimensioning
[ILP]	Integer Linear Programming
[IP]	Internet Protocol
[JET]	Just-Enough-Time
[LCG]	LHC Computing Grid
[LHC]	Large Hadron Collider
[LMS]	Least Mean Squares
[MEMS]	Micro Electro Mechanical Systems
[MG]	Multi-Granular
[MG-OXC]	Multi-Granular Optical Cross-Connect
[NE]	Network Element
[NEV]	Network Element Vector

[NP]	Nondeterministic Polynomial
[NREN]	National Research and Education Network
[OBS]	Optical Burst Switching
[OBS]	Optical Burst Switching
[OCP]	Optical Control Plane
[OCS]	Optical Circuit Switching
[OPS]	Optical Packet Switching
[OPS]	Optical Packet Switching
[OXC]	Optical Cross-Connect
[PTS]	Data Packet Transmission Time
[Q3A]	QoS Aware Anycasting Algorithm
[QoS]	Quality of Service
[SCS]	Spectral Clustering Scheduling
[SOA]	Semiconductor Optical Amplifier
[SP]	Shortest Path
[SPT]	Shortest Path Tree
[SPM]	Self-Phase Modulation
[TE]	Traffic Engineering
[TED]	Traffic Engineering Database
[WCC]	Wavelength-Continuity Constraint
[WDM]	Wavelength Division Multiplexing
[WR-OBS]	Wavelength-Routed Optical Burst Switched Networks
[XPM]	Cross-Phase Modulation

Appendix A Optical Parameter Values

A.1 Paremeter Values for SMF and DCF

The parameters for the two fibre types (DCF and SMF) used in our link model are shown in **Błąd! Nie można odnaleźć źródła odwołania.**.

Parameter Name	SMF (Single Mode Fibre)	DCF (Dispersion Compensation Fibre)
Attenuation a (dB/km)	0.25	0.5
Nonlinear index coefficient (m^2/W)	$2.6*10^{-20}$	$3.5*10^{-20}$
Chromatic Dispersion Parameter (s/m^2)	$17*10^{-6}$	$-80*10^{-6}$
Dispersion Slope (s/m)	$0.085*10^{3}$	$-0.3*10^{-3}$
Effective Area (m^2)	$65*10^{-12}$	$22*10^{-12}$

Table 8: Parameter Values of Fibre Types

Appendix B ILP Formulations

B.1 ILP Formulation of G-ND Problem

Input: Topology G=(V,E)

Link lengths $l : E \rightarrow □$

Demands $h_d (d = 1..D)$

Number of alternative shortest paths $k \in \mathbb{Z}_+$

Set P of candidate paths

$$\delta_{edp} = \begin{cases} 1, \textit{if link e is used by path p to serve demand d} \\ 0, \textit{otherwise} \end{cases} , e = 1..|E|, d = 1..D, p = 1..k$$

$$\eta_{e,n} = \begin{cases} 1, \textit{if link e is incident to node n} \\ 0, \textit{otherwise} \end{cases} , e = 1..|E|, n = 1..|V|)$$

Maximum number of wavelengths per fiber *W*

Trenching cost *α* per link length unit

Trenching cost α_e of link e: $\alpha_e = \alpha \cdot l(e)$

Span cost $\beta_{span} \in □$

Fixed fibre cost $\beta_{fix} \in \mathbb{Q}$

Span length $l_{span} \in \mathbb{Q}$

Fiber cost β_e of link e: $\beta_e = \frac{l(e)}{l_{span}} \cdot \beta_{span} + \beta_{fix} , e = 1..|E|$

Wavelength cost *γ*

Dimensions θ_s of switch of type *s*, $s \in S \, and \, S \subseteq □_+$

Cost φ_s per input port of switch of type *s*

Variables:

- $x_{dpc} \in □_+$: Number of lightpaths that use the *pth* path serving demand *d* on wavelength *c*.

- $w_{ce} \in □_+$: Number of times wavelength *c* is used on link *e*.

- $y_e \in \square_{+}$: Number of fibers installed on link *e*.

- $u_e \in \{0,1\} = \begin{cases} 1, \textit{if link e is used in the dimensioned network} \\ 0, \textit{otherwise} \end{cases}$

- $t_{n,s} \in \{0,1\} = \begin{cases} 1, \textit{if node n requires switch fabric dimensions corresponding to switch type s} \\ 0, \textit{otherwise} \end{cases} ,n = 1..|V|, s = 1..S$

$u_e \in \{0,1\} = \begin{cases} 1, \textit{if link e is used in the dimensioned network} \\ 0, \textit{otherwise} \end{cases}$

<u>Objective:</u> Minimize $Z = \sum_{e=1}^{|E|}(u_e \cdot \alpha_e) + \sum_{e=1}^{|E|}(y_e \cdot \beta_e) + \sum_{e=1}^{|E|}\sum_{c=1}^{W}(w_{ce} \cdot \gamma) + \sum_{n=1}^{|V|}\sum_{s=1}^{S}(t_{n,s} \cdot \varphi_s \cdot \theta_s \cdot W)$

<u>subject to:</u> $\sum_{p=1}^{K}\sum_{c=1}^{W} x_{dpc} \geq h_d \ ,d = 1..D$

$$y_e \geq w_{ce} \ ,c = 1..W, e = 1..|E|$$

$$\sum_{d=1}^{D}\sum_{p=1}^{K}(\delta_{edp} \cdot x_{dpc}) \leq w_{ce} ,c = 1..W, e = 1..|E|$$

$$y_e \leq u_e \cdot max\,fibre \ ,e = 1..|E|$$

$$\sum_{s=1}^{S} t_{n,s} = 1 \, , n = 1..|V|$$

$$\sum_{e=1}^{|E|}(2 \cdot \eta_{e,n} \cdot y_e) \leq \sum_{s=1}^{S} \theta_s \cdot t_{n,s} \ ,n = 1..|V|$$