Future Generation Computer Systems 26 (2010) 576–589

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Joint resource and network scheduling with adaptive offset determination for optical burst switched grids

Mehmet Koseoglu *, Ezhan Karasan

Department of Electrical and Electronics Engineering, Bilkent University, TR-06800, Bilkent, Ankara, Turkey

ARTICLE INFO

Article history:
Received 26 February 2009
Received in revised form
3 November 2009
Accepted 16 November 2009
Available online 24 November 2009

Keywords:
Grid computing
Optical burst switching
Grid resource provisioning
Grid network provisioning

ABSTRACT

Optical burst switching (OBS) is a promising technology for optical grids with short-lived and interactive data communication requirements. On the other hand, burst losses are in the nature of the OBS protocol and these losses severely affect the grid job completion times. This paper first proposes a joint grid resource and network provisioning method to avoid congestion in the network in order to minimize grid job completion times. Simulations show that joint provisioning significantly reduces completion times in comparison to other methods that perform network provisioning after grid scheduling. An adaptive extra offset based quality of service (QoS) mechanism is also proposed in order to reduce grid burst losses in case of network congestion. Results show that this adaptive mechanism significantly reduces grid completion times by exploiting the trade-off between decreasing loss probability and increasing delay introduced by the extra offset time.

1. Introduction

Computational requirements of scientific problems are growing dramatically and this increase requires collaboration between remote institutions in order to solve these problems. For example, the Large Hadron Collider at CERN will produce 15 petabytes of data in a year which has to be processed to determine useful observations [1]. Since it is not possible to process this enormous amount of data at a single processing site, geographically distributed resources have to collaborate to solve such problems. Grid computing introduces a new paradigm in which independent remote institutions collaborate through a uniform interface to solve complex problems [2].

In addition to scientific problems, grid computing is envisioned to be used for consumer applications in the future [3]. In this case, consumers rent computational resources from remote servers and pay as they get service. Consumer grids are expected to lower the initial and maintenance costs of expansive resources that may be required for running computationally expensive applications. Real-time rendering for video games or interactive high-definition TV are possible applications for this type of grid. These applications also require high bandwidth similar to science grids but the network infrastructure has to support more interactive traffic, and delay must be kept low for supporting real-time applications.

Although the natures of these two usages of grid computing are different, optical networks provide a suitable infrastructure for both [4]. In the scientific grid, the amount of data transferred is very high but the frequency of these transfers is low. On the other hand, the duration of data transfers is much smaller in the consumer grid but the interactions among entities are more frequent. For that reason, both applications require high bandwidth, which can be provided by optical networks.

Although optical networking is suitable for both applications, the switching method to be used in the optical grid has to be selected based on the specific grid application. For example, wavelength switching is more appropriate for data transfers that are long-lasting and whose bandwidth demands are not fluctuating. Meanwhile, optical burst switching (OBS) performs better when the data traffic is short-lived and dynamic [5]. Hence, wavelength switching is more suitable for science grids whereas OBS is more suitable for consumer grids.

Even though OBS is suitable for consumer grids, burst losses have to be considered when running the grid over OBS networks. In the OBS protocol, a control packet is sent before the data burst in order to reserve network resources. The data burst is sent after a predetermined duration without waiting for an acknowledgment so that the delay is kept at a minimum. Because of this one-way reservation mechanism, a data burst can be lost if its control packet could not reserve network resources. Lost bursts carrying grid jobs need to be retransmitted, resulting in an increase in the completion times of grid jobs.

The consumers in a grid environment have flexibility in both resource selection and network path selection. Since the consumers in a grid environment can request service from various providers,

* Corresponding author.
E-mail addresses: kmehmet@ee.bilkent.edu.tr (M. Koseoglu), ezhan@ee.bilkent.edu.tr (E. Karasan).

0167-739X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2009.11.002

they can perform resource and network scheduling jointly to improve performance. There are several proposals describing how joint scheduling can be done for optical grids running over wavelength routed networks [6,7]. Our study investigates the joint resource and network scheduling problem in OBS grid networks. In the OBS grid, if the paths to a resource which offers a short processing delay are severely congested, a consumer may decide to select another resource with slower processing but with less congested paths. Beside resource selection, a consumer also has the option to select among the paths destined to the resource to send the grid job with less delay. We show that joint selection of computational resources and network paths reduces grid job completion times significantly by lowering loss rates when the transmission times are comparable with the grid computation times, which is the case for consumer grids.

In this paper, we first propose a joint resource and path selection algorithm and it is shown to outperform other algorithms that make resource and path selection separately. We then extend this scheme with the adaptive extra offset mechanism which adaptively selects the offset of the grid bursts depending on the congestion in the network. The extra offset mechanism for OBS bursts decreases the loss probability of bursts at the expense of increasing the latency. Since the completion time of an OBS grid job is a function of both loss probability and delay, reducing the loss probability of a burst using extra offset requires fewer retransmissions, so the average completion time may decrease although the transmission delay is increased by the extra offset. The proposed mechanism finds the optimum extra offset which minimizes the average completion time by exploiting this tradeoff between the delay and loss probability.

We improve the previous version of this work [8] by extending the numerical results and by providing several mathematical analyses. The numerical results are extended to include the effects of changing best-effort traffic load and burstiness as well as the effects of computational resource parameters on the average completion time. In addition to these, the control plane load created by the proposed mechanism is analyzed. We also provide a mathematical method for calculating the optimum extra offset.

The remainder of the paper is structured as follows. In the next section, we discuss how the OBS grid architecture addresses latency problems with the contemporary grid architecture. In Section 3, the optical grid architectures in the literature are presented. Analysis of a grid job lifetime is given in Section 4. In Sections 5 and 6, we present the proposed algorithms, joint resource-path selection and adaptive offset based QoS mechanism, respectively. The extra OBS control plane load generated by the proposed algorithms is analyzed in Section 7. Performance evaluation results for the algorithms are presented in Section 8.

2. The need for a low latency grid architecture

Since current grid implementations are generally designed for long-lasting jobs, the overhead caused by resource and network scheduling does not significantly affect the whole duration of the computation. However, as grid computing is starting to be used by highly interactive applications where the job lengths are shorter, these overheads become significant. For that reason, grid computing models have to be revisited to perform scheduling operations faster.

There are two types of models proposed for scheduling network and grid resources in wavelength switched grids [9]. The first is the overlay model, where applications request resource scheduling from the grid middleware. Once resource allocation is completed, applications ask for network connectivity to the selected resources. Network provisioning is also performed by the grid middleware, which communicates with the optical control plane to reserve end-to-end lightpaths between the user and the resource.

The second solution is to establish a unified control plane for both grid and network resource provisioning. A network layer protocol such as GMPLS [10] can be used for grid resource reservation as well as network provisioning. Delays caused by the middleware can be reduced to some extent by integrating the network and resource reservations. However, since wavelength switching requires connection set-up, this approach is still not suitable for small data transfers.

The OBS grid architecture is proposed for addressing the delay problems in the current grid practices and it makes the grid more suitable for small-sized real-time jobs. Currently, grid resource allocation is done in a centralized manner. However, centralized allocation is not feasible for an increasing number of users with highly dynamic requests [3]. For that reason, OBS grid architecture enables a distributed way of resource reservation where users interact directly with resources. In addition to distributed resource reservation, low latency data transmission provided by the OBS protocol makes this architecture suitable for dynamic jobs.

3. OBS grid architectures

In this section, the consumer grid architecture based on OBS is discussed. OBS offers sub-wavelength granularity for optical networks [11]. In OBS, a burst control packet (BCP) is sent before the optical data to configure switches between the source and the destination. The optical burst is delayed at the source node for an offset time waiting for the control packet to configure an all-optical path and the BCP is sent without waiting for an acknowledgment. If the BCP fails to find an available wavelength for a link, the optical burst is dropped at that node.

OBS performs better than wavelength switching for grids with dynamic and short-lived data transmission requirements. Since each lightpath has a bandwidth of one wavelength, the granularity of wavelength switching is very coarse. Besides, setting up a lightpath takes in the order of hundred milliseconds so it is not suitable for short-lived connections. As shown in [5], OBS performs better than wavelength switching for grids with scarce wavelength resources.

There are several proposals for consumer grid architectures running over OBS networks [12,13]. In a consumer grid, jobs are small in contrast to science grids, and assigning a single job to multiple resources increases the communication time overhead. For that reason, a grid networking architecture where each job is assigned to a single resource is considered, as in [14]. In the OBS grid, an optical burst carries a single grid job. The BCP carries the grid job information which is sent to the network without a specific destination address (anycasting). The BCP is routed to a suitable grid resource by intelligent routers which have grid layer information in addition to network layer information. After the offset time, the burst containing the grid job is sent to the network without waiting for an acknowledgment.

Instead of leaving the resource selection to intelligent routers in the network, another reservation mechanism is presented in [13], where the resource provisioning and wavelength reservation are performed in two separate steps. In this mechanism, which is called explicit reservation, a BCP containing the job description is sent to the network for resource discovery. After receiving these discovery probes, intermediate routers perform resource discovery and send acknowledgments to the consumer if they find a suitable resource. Fully aware of all options, the consumer selects a resource and sends the job as an optical burst to the selected resource.

In our work, we use explicit reservation architecture because we find this architecture more practical. In the case of anycasting, the routers need to perform on-the-fly routing based on resource states, and resource information has to be distributed to routers in the network. This approach significantly increases the complexity of routers and the cost of the infrastructure. Instead, we use a

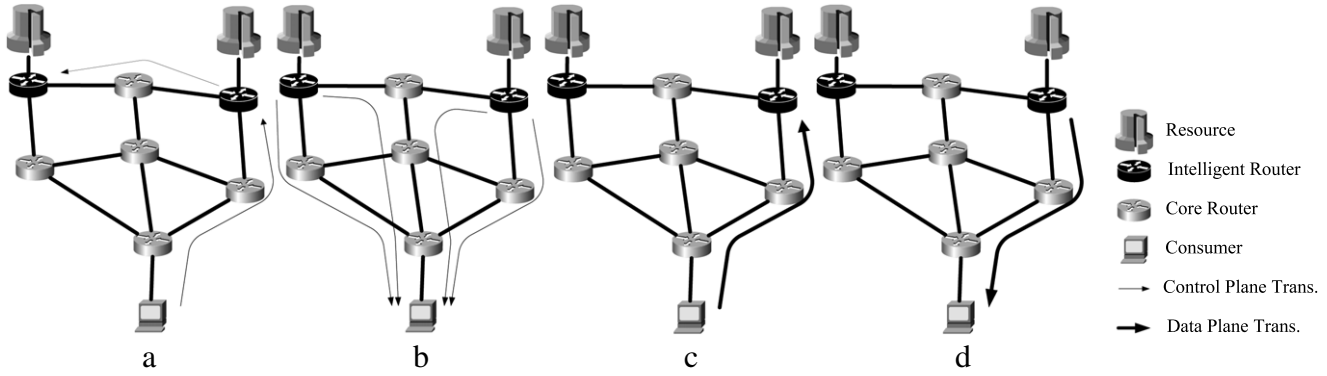

Fig. 1. (a) The consumer sends the job specification to the nearest intelligent router using the OBS control plane, and this router multicasts this specification to other intelligent routers in the network. (b) Resources send acknowledgment packets back to the consumer and the core routers piggyback load information to these acknowledgment packets on their way back. (c) After the consumer receives the resource information and network load information from the acknowledgment packets, it decides on one of the resources and the path to that resource and sends the job in the form of an optical burst. (d) After the resource completes the processing of the job, it sends the results in the form of an optical burst.

fully consumer-controlled architecture in which both grid resource provisioning and route selection are performed by the source of the job. Since the full path of a grid burst is determined by the consumer, the core routers in our architecture do not make any routing decisions; as such they only forward the bursts according to the path specified in the BCPs. However, routers near a resource can query that resource and send information to the consumers regarding the resource. These routers are called intelligent routers.

In the architecture we study, consumer-controlled resource discovery is achieved by multicasting a discovery message to all available resources in the grid using the OBS control plane, as shown in Fig. 1. This discovery message which includes job specifications is sent to the nearest intelligent router and it is multicasted to other intelligent routers by this router. The resources respond to these discovery messages by reserving their processors for the mentioned job and by sending the expected completion time using resource offering packets that are transmitted over the OBS control plane, as shown in Fig. 1(b). Resource reservations start at the expected arrival time of a job instead of starting immediately after the reception of the resource offering packets in order to increase utilization. Resource offering packets are sent to the consumer using two disjoint paths for feedback collection which will be explained later.

The first mechanism integrated to the OBS grid architecture is the joint path and resource selection mechanism. This mechanism is a generalization of the path switching methods proposed in the literature. Path switching maintains a set of alternate paths between each source–destination pair and ranks the paths based on their recent congestion levels. Path switching was first proposed for IP traffic in order to take advantage of diverse path availability of the Internet. [15,16] propose heuristic mechanisms for path switching in an OBS network and show that reduced burst loss rates can be achieved by path switching.

In our path switching mechanism, we limit the number of alternate paths to two for the following two reasons. First, since discovery packets are sent through each alternate path using the control plane, the load on the control plane is increased with every alternate path. Also, various studies show that the performance improvement obtained by using more than two paths is not significant when alternate routing is used. For WDM networks, it is shown in [17,18] that, when the number of alternate paths is increased from one to two, the performance improvement is significant. However, increasing the number of alternate paths further does not provide a significant improvement. In another study on OBS networks [19], it is shown that the loss probability increases when more than two alternate paths are used because as the number of alternate paths increases, the average lengths of these paths also increase, making the burst losses more likely due to path length priority effect of OBS [20].

Since the source has to collect information about different paths to a destination in order to perform source-controlled path switching, a mechanism for collecting path statistics is integrated. The core OBS routers record the average offered traffic load and piggyback this information into the resource offering packets sent by the resources. The reason that the resource offering packets are sent over two disjoint paths is to collect the load information of two disjoint paths for each resource.

After the consumer receives the resource offering packets carrying the completion time offerings of resources and load information of the paths, it selects the resource to execute the job and path to that resource using our proposed algorithm. Then, the consumer sends the job over the selected path in form of an optical burst preceded by a burst control packet, which is shown in Fig. 1(c). The other resources cancel their reservations after a predetermined duration when they do not receive the job burst.

We also employ an acknowledgment mechanism for explicit burst loss notification because without explicit acknowledgments loss of a burst can only be noticed by the absence of the result burst. Since that would significantly increase the completion time of a job, resources send acknowledgment packets using the OBS control plane when they receive job bursts.

We also employ the same path switching mechanism on the resource side but the resources need to probe the network actively to collect feedback in this case. The resource sends probe packets to the consumer over the two link-disjoint paths and the consumer sends these packets back to the resource using the paths on they are arriving. Core routers write their offered load information to these packets so that the resource acquires the load information over both paths. Then, the resource selects the path to send the result to the consumer and transmits it using the standard OBS protocol, as shown in Fig. 1(d).

We also propose a service differentiation mechanism in order to give priority to grid bursts in case there is a best-effort traffic which shares the network with the grid traffic. The proposed service differentiation mechanism is based on the offset-induced priority where an extra offset is applied to high priority bursts [21]. The extra offset increases the successful reservation probability of a burst but it also increases the delay. Our mechanism determines the optimum offset time which minimizes the completion time of the grid job.

4. Analysis of the completion time of a grid job

The completion time of a grid job consists of several components whose lengths depend on several factors such as state of the resources, network congestion and network size. We analyze the completion time of the grid job in this section to develop the proposed methods for completion time minimization. Since our proposed mechanisms work both at the consumer side and the resource side, we present both problems that the consumer and the resource need to solve separately.

4.1. Analysis from the consumer point of view

The completion time of a grid job can be separated into three major components. The first component is the discovery time, which covers the dissemination of the job specification and collection of resource offering packets. This duration is dependent on the size of the network and the speed of resource querying performed by routers. The second component, resource processing time, is the time spent at the computational resource, which includes queuing delay and job processing time. The queuing delay depends on the level of congestion at the resource, and the processing time depends on the amount of computation required by the job. The third term of the completion time is the networking time, which includes transmission and propagation delays of optical bursts as well as retransmissions of lost bursts.

We use the following notation in order to express the completion time first in the case of no burst losses. We will then handle the case with burst losses.

- τ_{dis}: Discovery time
- Δ_j: Offset time of the job burst
- τ_{jl}: Transmission time of the job burst
- τ_{jp}: Propagation delay of the job burst
- τ_{res}: Resource time
- Δ_r: Offset time of the result burst
- τ_{rl}: Transmission time of the result burst
- τ_{rp}: Propagation delay of the result burst

As the timeline for the no-loss scenario shown in Fig. 2(a) is examined, it can be observed that the total job completion time, T_{min}, is given by

$$T_{\text{min}} = \tau_{\text{dis}} + \tau_{jl} + \tau_{jp} + \tau_{\text{res}} + \tau_{rl} + \tau_{rp} + \Delta_j + \Delta_r.$$

If we assume that the transmission times and propagation delays of job and result bursts are same, that is, $\tau_l = \tau_{jl} = \tau_{rl}$ and $\tau_p = \tau_{jp} = \tau_{rp}$, the expression simplifies to

$$T_{\text{min}} = \tau_{\text{dis}} + 2\tau_l + 2\tau_p + \tau_{\text{res}} + \Delta_j + \Delta_r. \quad (1)$$

If the job burst is lost, resource discovery has to be performed again and the job burst has to be sent to the newly selected resource. The repetition of the resource discovery is required because the resources clear processor reservations when a job does not arrive on time.

A burst loss can be noticed by the consumer when the acknowledgment packet does not arrive in a predetermined timeout duration, T_t. Timeout duration includes the propagation and transmission delays of the burst and propagation delay of the acknowledgment. It also includes a guard band, τ_g, for unexpected delays such as processing or transmission delays which may occur at the consumer side. Since we assume that the propagation delay of the job burst is equal to the propagation delay of the acknowledgment, the timeout duration is given by

$$T_t = \tau_l + 2\tau_p + \tau_g + \Delta_j.$$

The timeline of the grid job when the job burst is lost can be seen in Fig. 2(b). From this figure, it can be observed that the retransmission cost associated with a job burst loss is given by

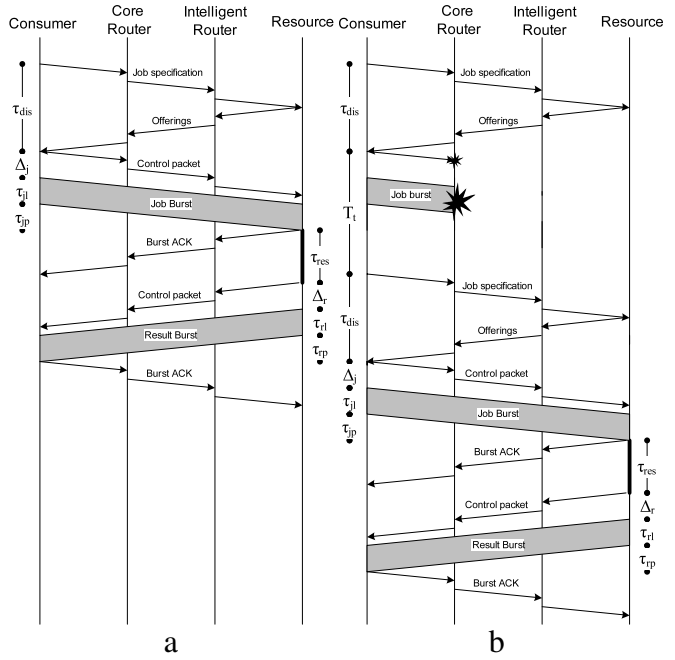

Fig. 2. Timeline of a grid job (a) when there is no burst loss and (b) when the job burst is lost.

$$\begin{aligned} T_{rt} &= T_t + \tau_{\text{dis}} \\ &= \tau_l + 2\tau_p + \tau_g + \tau_{\text{dis}} + \Delta_j. \end{aligned}$$

We formulate an expected completion expression using these expressions. Let $P_p^{(n)}$ be the loss probability of the grid job burst and $T_{rt}^{(n)}$ be the retransmission cost in the nth transmission attempt; T_{min} is given by (1). Then the expected completion time can be written as follows:

$$E[T_{\text{total}}] = T_{\text{min}} + \sum_{i=1}^{\infty} \left(\prod_{j=1}^{i} P_p^{(j)} \right) T_{rt}^{(i)}.$$

Assuming that the network and computational resource conditions do not change between transmission attempts, we have $P_p^{(n)} = P_p$ and $T_{rt}^{(n)} = T_{rt}$. Then, the expected completion time of a grid job can be expressed as

$$E[T_{\text{total}}] = T_{\text{min}} + T_{rt} \frac{P_p}{1 - P_p}. \quad (2)$$

Using the completion time as the single metric of resource and path selection, we aim to reduce the completion times of grid jobs by computing the expected completion time of a job when a specific resource and path combination is selected.

4.2. Analysis from the resource point of view

When the job is processed at the computational resource, the results have to be transmitted to the consumer that created the job. The timelines of bursts carrying the results are shown in Fig. 3(a) for the no-loss case and in Fig. 3(b) for the loss case. Result bursts are delivered quicker than job bursts since there is no discovery time or processing time.

After the processing, the minimum required time to complete the job can be written as

$$T_{\text{min}} = \tau_{rl} + \tau_{rp} + \Delta_r. \quad (3)$$

If the job burst is lost, the retransmission cost becomes the timeout duration, which is required to notice the loss of the burst in addition to a guard band, as given by

$$T_{rt} = \tau_{rl} + \tau_{rp} + \tau_g + \Delta_r. \quad (4)$$

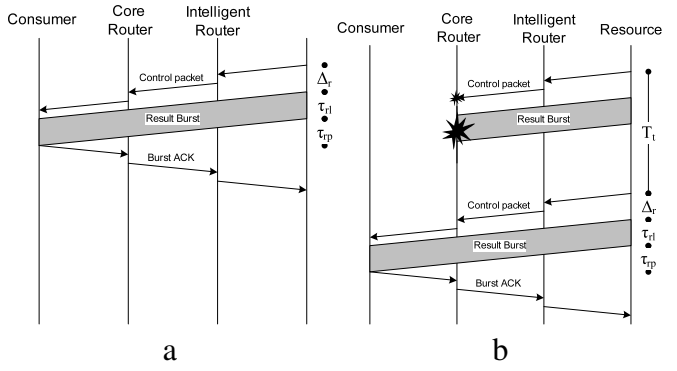

Fig. 3. Timeline of a result burst (a) when there is no burst loss and (b) when the result burst is lost.

In contrast to the job burst whose destination can be selected among several alternative resources, there is no such flexibility for result bursts. The destination of the result burst is the consumer which creates the job, and resources only perform path selection.

In the next section, the joint resource and path selection algorithm performed by the consumer and the path selection algorithm performed by the resource are explained.

5. Resource and path selection

In this section, we use the analysis presented in the previous section to develop resource and path selection algorithms for the consumer and resource.

5.1. Joint resource and path selection for consumer

This algorithm is based on the estimation of the expected completion time function given by (2). To estimate the expected completion time, the consumer has to estimate all components of the completion time presented in Section 4.1. We assume that the discovery time is fixed. To estimate the resource processing time, we use the resource offering messages coming from each resource. For the network time estimation, we assume that the propagation delays to each resource are known by the consumer and the required transmission offset times due to OBS control plane processing of all bursts are negligible. The transmission time of the job burst is determined by the data size of the job and the line speed.

In addition to these components, loss probability estimation is also required for the computation of the expected completion time. For this estimation, load information of the links that are piggybacked to the resource offering packets is used. These load levels are recorded by each core router by calculating the total duration of bursts offered to its links during a predefined time window. The load level for each link in Erlangs is expressed as

$$A_l = \frac{\Lambda_{of}}{\Lambda_{\text{win}}}$$

where Λ_{of} is the total duration of bursts offered to this link during the time window, Λ_{win}. At the end of each time window, these statistics are reset to zero. In calculating the burst loss probability, the consumer should also consider the increase of traffic load over a link when it starts to route its whole traffic over a new path using that link. This correction is especially important when the load on a link is generated by a few consumers. The estimated traffic load on link l after rerouting is given by

$$A_{le} = A_l + \delta$$

where δ is the difference of the traffic load on link l generated by the consumer after rerouting. Then, the loss rate on link l, π_l, can be computed using the Erlang-B formula given by

$$\pi_l = \frac{\frac{A_{le}^W}{W!}}{\sum_{i=0}^{W} \frac{A_{le}^i}{i!}}.$$

Using the link independence assumption, the loss probability over path p can be expressed as

$$P_p = 1 - \prod_{l \in p} (1 - \pi_l). \tag{5}$$

Using the estimations of time components and loss probabilities for each resource and path, the consumer computes the expected completion time function for each resource–path pair and selects the pair which results in the minimum expected completion time:

$$E[T^{rp}] = T^{rp}_{\min} + T^{rp}_{rt} \frac{P_p}{1 - P_p}$$

where

$$T^{rp}_{\min} = \tau_{\text{dis}} + 2\tau_l + 2\tau^{rp}_p + \tau^{r}_{\text{res}}$$

and

$$T^{rp}_{rt} = T^{rp}_t + \tau_{\text{dis}}.$$

Since all sources implement their own path switching algorithm independent of each other, the grid traffic may oscillate between alternate paths if sources make similar path switching decisions in a synchronized fashion. In this case, more than one traffic source may select paths which use the same underutilized links, making those links congested. After receiving load reports in the next time window, all of these sources switch away from those links. These oscillations continue when all sources return to their first choices in the next time window. For that reason, we implement a threshold based hysteresis mechanism to prevent this kind of oscillation. In this mechanism, a source does not switch its resource and path choice in the previous time window unless more than 10% improvement obtained in estimated completion time.

5.2. Path selection for resource

Similar to the consumer, the resource evaluates the expected completion time function using the estimations of (3), (4) and (5). In this case, the resource evaluates this function for only two disjoint paths to the consumer since the destination of the result is fixed. It selects the path which results in lower expected transmission time.

In the next section, we present the proposed adaptive extra offset determination mechanism which minimizes the job completion time.

6. Adaptive QoS offset determination

When the OBS network infrastructure of the grid is also used for carrying best-effort traffic, a mechanism for service differentiation between grid and best-effort traffic is needed since the grid application is delay sensitive. In this section, we propose a service differentiation method for grid traffic based on the extra offset based service differentiation. Our method adaptively determines the extra offset value of bursts depending on the congestion in the network, minimizing the completion time of grid jobs.

6.1. Analysis of effect of QoS offset on completion time

The increase in the offset time of a burst results in the increase of the minimum completion time, as can be seen from (1). However, it is possible to reduce the expected completion time function by reducing the loss probability of a burst, as can be seen in (2). Here, we analyze the change in completion time with changing QoS offset.

We use the mathematical model given in [22] to perform this analysis. In this model, there are two classes of traffic, one of which is sent using a QoS offset. Let A_l^G and A_l^B denote the grid traffic load and the best-effort traffic load on link l, respectively. Assuming that the grid traffic constitutes the high priority class, the loss probability of grid traffic can be written as follows:

$$\pi_l^G = B(A_l^G + Y^B(\Delta), W) \tag{6}$$

where $Y^B(\Delta)$ is the low priority best-effort traffic which is seen by the grid traffic with a QoS offset of Δ. $Y^B(\Delta)$ can be written as follows:

$$Y^B(\Delta) = A_l^B(1-\pi_l^B)(1-R^B(\Delta)) \tag{7}$$

where π_l^B is the loss probability of the bursts belonging to the best-effort traffic over link l and R^B is the excess life distribution function of the best-effort bursts. Excess life distribution is the distribution of the residual duration of a burst after a random point in time. R^B is written as

$$R^B(\Delta) = \frac{1}{h^B}\int_0^{\Delta}(1-F^B(u))\mathrm{d}u \tag{8}$$

where $F^B(u)$ is the burst length distribution of best-effort bursts and h^B is the mean of best-effort bursts. After the computation of the grid loss probability, the loss probability can be approximated by the conservation law

$$A_l\pi_l = A_l^G\pi_l^G + A_l^B\pi_l^B. \tag{9}$$

Since the loss probabilities of the grid and best-effort traffic are interdependent, these equations have to be solved iteratively, as described in [22].

6.2. Analysis of optimum offset time

In this section, we present the mathematical derivation of the optimum offset value for a grid job burst. To analyze the effect of the job burst offset we rewrite (2) as follows:

$$\begin{aligned}E[T_{\text{total}}] &= (T_{\min} - T_{rt}) + \frac{T_{rt}}{1-P_p(\Delta_j)}\\ &= T_0 + \frac{T_1+\Delta_j}{1-P_p(\Delta_j)}\end{aligned}$$

where $T_0 = T_{\min} - T_{rt}$ and $T_1 = T_{rt} - \Delta_j$. To find the optimum offset, we need to take the derivative of the completion time with respect to Δ_j and equate to 0:

$$\frac{\partial T_{\text{total}}}{\partial \Delta_j} = \frac{1}{1-P_p(\Delta_j)}\left(1 + \frac{T_1+\Delta_j}{1-P_p(\Delta_j)}\frac{\partial P_p}{\partial \Delta_j}\right). \tag{10}$$

Since the link loss probabilities are dependent on Δ_j, (5) can be rewritten as

$$P_p(\Delta_j) = 1 - \prod_{l\in p}(1-\pi_l(\Delta_j)).$$

The derivative of this expression is given by

$$\frac{\partial P_p}{\partial \Delta_j} = (1-P_p(\Delta_j))\sum_{l\in p}\frac{\pi_l'(\Delta_j)}{1-\pi_l(\Delta_j)} \tag{11}$$

which can be used to write (10) as

$$\begin{aligned}\frac{\partial T_{\text{total}}}{\partial \Delta_j} &= \left(1 + (\Delta_j + T_1)\sum_{l\in p}\frac{\pi_l'(\Delta_j)}{1-\pi_l(\Delta_j)}\right)\\ &\quad\cdot\frac{1}{1-P_p(\Delta_j)}.\end{aligned} \tag{12}$$

The grid burst loss probability $\pi_l(\Delta_j)$ can be computed by performing fixed-point iterations using (6)–(9). However, to simplify this analysis we assume that the low priority link loss probability, π_l^B, is independent of Δ_j, and we call it q_l^B. So, the grid burst loss probability becomes

$$\pi_l(\Delta_j) = B(\rho_l, W_l)$$

where $\rho_l = A_l^G + A_l^B(1-q_l^B)(1-R^B(\Delta_j))$, W_l is the number of wavelengths at link l and $B(\cdot,\cdot)$ is the Erlang-B formula. Using the following identity for the derivative of the Erlang-B formula [23]

$$\frac{\partial}{\partial\rho}B(\rho,W) = B(\rho,W)\frac{W-\rho(1-B(\rho,W))}{\rho}$$

we can write

$$\pi_l'(\Delta_j) = \pi_l(\Delta_j)\left(\frac{W_l}{\rho_l} - (1-\pi_l(\Delta_j))\right)\frac{\partial\rho_l}{\partial\Delta_j}$$

where

$$\frac{\partial\rho_l}{\partial\Delta_j} = -\frac{1}{h_B}(1-F(\Delta_j))A_l^B(1-q_l^B).$$

Then, the term in the summation in (11) can be written as

$$\begin{aligned}\frac{\pi_l'(\Delta_j)}{1-\pi_l(\Delta_j)} &= \frac{1}{h_B}(1-F(\Delta_j))\pi_l(\Delta_j)A_l^B\\ &\quad\cdot(1-q_l^B)\left(1-\frac{W_l}{(1-\pi_l(\Delta_j))\rho_l}\right).\end{aligned} \tag{13}$$

When we equate (10) to zero, we get

$$\Delta_j = -\left(\sum_{l\in p}\frac{\pi_l'(\Delta_j)}{1-\pi_l(\Delta_j)}\right)^{-1} - T_1. \tag{14}$$

(14) can be solved using (13) and fixed-point iterations. Starting with a predetermined value, Δ is updated according to

$$\Delta_j \leftarrow \max\left(-\left(\sum_{l\in p}\frac{\pi_l'(\Delta_j)}{1-\pi_l(\Delta_j)}\right)^{-1} - T_1, 0\right)$$

since the offset value cannot be smaller than 0. The existence of a fixed point follows from Brouwer's fixed-point theorem, which states that, if $C \in \mathbb{R}^n$ is a closed, bounded and convex set, then a continuous function $f : C \to C$ has a fixed point in C. Since the offset can only take values between 0 and the maximum low priority burst length including the boundaries, the required conditions are satisfied. Also, the function is continuous except if $\pi_l = 0$ for all l or $\pi_l = 1$ for any l. Since these conditions are never satisfied, the function has a fixed point.

6.3. Complexity analysis

The complexity of the Erlang-B formula is $O(W)$. According to (14), Erlang-B formula has to be evaluated for each link on the disjoint paths to each resource. The total number of Erlang-B calculations for each iteration is equal to $2N_{dp}N_r$, where N_{dp} is the average hop number of disjoint paths and N_r is the number of resources. This computation has to be performed for multiple times such that the convergence to optimum offset is achieved. The overall complexity of the optimum offset time computation is given by $O(N_{dp}N_rWK)$, where K is the number of iterations until the fixed-point iterations converge.

7. Analysis of OBS control plane load

In this section, we quantify the extra control plane load generated in order to collect the feedback from the network that is necessary for implementing the proposed scheduling mechanisms. The

extra control packets transmitted for implementing the proposed mechanisms are the duplicate offering packets sent in the discovery phase and probe packets sent by the resources before completion of the processing.

In order to evaluate this extra load, a definition is needed for the control plane load. A reasonable definition for the load created by a single control packet is the product of the number of hops that are transmitted and the processing cost of that packet at each hop. Although it is not possible to exactly measure the processing cost associated with each type of control packet, it is useful to distinguish between the control packets that involve wavelength scheduling and switch configuration and other control packets routed without any configuration. The burst control packets sent for the job bursts and the result bursts will require longer processing times than the packets sent for resource discovery, burst acknowledgments and probe packets because the latter are forwarded without communicating with the switch control unit. If we define C_f as the processing cost associated with forwarding of a control packet and C_s as the processing cost of wavelength scheduling and configuration of the switch, the cost of processing for burst control packets becomes $C_f + C_s$ while the cost for other control packets is only C_f.

The average control plane load created by a single grid job can be expressed as the sum of loads of its individual phases:

$$\lambda_{\text{total}} = \lambda_{\text{dis}} + \lambda_{\text{job}} + \lambda_{\text{ack}} + \lambda_{\text{probe}} + \lambda_{\text{result}}$$

where λ_{dis}, λ_{job}, λ_{ack}, λ_{probe} and λ_{result} correspond to the average load caused by resource discovery, job burst control, burst acknowledgment, probe and result burst control packets, respectively.

Let us define N_{dp} and N_{sp} as the average hop numbers of the disjoint paths and shortest paths between the consumers and resources, respectively. Since the job burst control packets and result burst control packets are sent using one of the disjoint paths between the consumer and the resource, $\lambda_{\text{job}} = \lambda_{\text{result}} = N_{dp}(C_f + C_s)$. Burst acknowledgment packets are also sent over one of the disjoint paths but they do not perform switch configuration so $\lambda_{\text{ack}} = N_{dp}C_f$. On the other hand, when a resource sends a probe packet to the consumer, it is sent over two paths and also bounced back by the consumer, so $\lambda_{\text{probe}} = 4N_{dp}C_f$.

The load generated by resource discovery is the sum of the loads created by the multicasting of the job specification and by the transmission of the resource offering packets from intelligent routers to consumers. The first component, the load created by the job specification dissemination, is equal to the size of the multicast tree connecting the intelligent routers multiplied by the processing cost, $N_{mult}C_f$. The second component is the load generated by offering packets sent over two disjoint paths. Since every intelligent router in the grid sends two offering packets to a consumer, λ_{dis} is expressed as

$$\lambda_{\text{dis}} = (N_{mult} + 2IN_{dp})C_f$$

where I is the number of intelligent resources. We can write the total load created by a single grid as

$$\lambda_f = ((2I + 7)N_{dp} + N_{mult})C_f + 2N_{dp}C_s.$$

If the proposed feedback collection mechanism were not used, the overhead created by the probe packets and the duplicate resource offering packet would not exist. Also, if we assume that shortest path routing is employed, the average hop number between the consumer and a resource is decreased. Then, the load without feedback mechanisms is given by

$$\lambda_{nf} = ((I + 3)N_{sp} + N_{mult})C_f + 2N_{sp}C_s.$$

Then the incremental load, U, necessary for implementing the proposed mechanisms can be written as

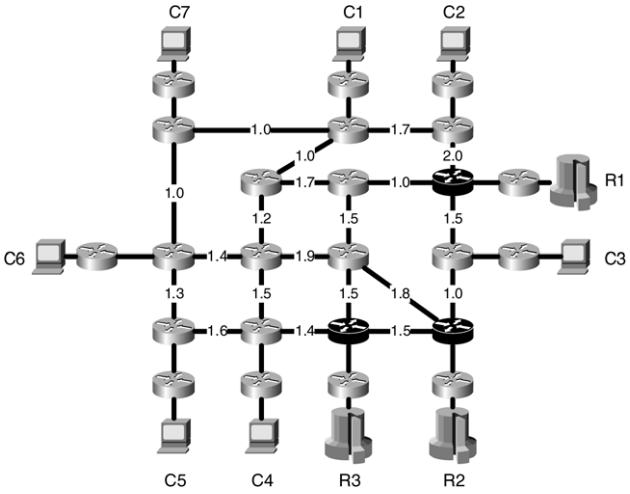

Fig. 4. The simulated OBS grid topology. The numbers show the propagation delay of each link in ms.

$$U = \frac{\lambda_f - \lambda_{nf}}{\lambda_{nf}} = \frac{((2I + 7)N_{dp} - (I + 3)N_{sp})C_f}{((I + 3)N_{sp} + N_{mult})C_f + 2N_{dp}C_s} + \frac{2(N_{dp} - N_{sp})C_s}{((I + 3)N_{sp} + N_{mult})C_f + 2N_{dp}C_s}. \quad (15)$$

The expression for the incremental load, U, given by (15) can be simplified to $U \approx \frac{N_{dp}-N_{sp}}{N_{sp}}$ if the assumption $C_s \gg C_f$ holds. This assumption seems reasonable because wavelength scheduling and switch configuration require the processing unit to communicate with the switching unit and configuration of the switch. For that reason, switch configuration and burst scheduling is expected to take longer than processing of a control packet.

8. Numerical results

8.1. Grid network model

The OBS grid network shown in Fig. 4 is used in simulations where the length of each core link is indicated. In this topology, there are seven customers and three resources. Each customer and resource is connected to the core network through an edge router. Also, the router adjacent to each resource is capable of querying the nearby resource and it sends acknowledgments to the consumer regarding that resource.

The length of the best-effort bursts and grid bursts is distributed uniformly between 0.5 ms and 15 ms. Each optical burst carries a single grid job or grid job result. We assume that the result of a job has the same data size as the job itself. The switching time for the core switches is 0.1 ms and control packet processing time is negligible. There are $W = 5$ wavelengths per fiber at each link and one of them is reserved for the OBS control plane. Also, we assume that there are ten links between the edge routers and the core network in order to prevent congestion at the edges of the network. The core routers record their load measurements periodically using $\Lambda_{\text{win}} = 1s$. Each simulation is performed for 300,000 jobs; however, only the statistics of the last 50,000 is taken into account in order to ensure that the simulations reach a stable state.

8.2. Computational resource model

In order to perform a realistic simulation of an OBS consumer grid, a workload model for grid jobs is required to generate the grid job parameters, to estimate the execution times of jobs and

to schedule jobs at the resources. There are studies which analyze the characteristics of the workload for some science grids [24], but since there is no consumer grid realized in practice, it is not possible to make use of these measurements.

Since we cannot find a workload trace for consumer grids, we use a speedup model from the literature for the execution of grid jobs on parallel processors. In a grid environment, computational resources have multiple processors and parts of the submitted jobs can be executed in parallel on these multiple processors. Depending on the characteristics of the job, the number of processors that will be used in execution may be fixed or variable. These days, most of the computational jobs in a grid are moldable jobs for which the number of processors that are used for execution can be determined by the executing resource [25]. This flexibility allows resources to schedule jobs according to the cost metric they choose.

We use Downey's model [26] to compute the execution times of jobs in our simulations. This model is used to estimate the speedup obtained parallel execution of grid jobs, which can be defined as

$$S_n = \frac{T(1)}{T(n)}$$

where $T(n)$ is the parallel runtime of a job on n processors and $T(1)$ is the sequential runtime of the same job. The speedup obtained by executing a job on multiple processors does not change linearly as the number of processors increases, and this fact affects the scheduling decisions made by the resource. Downey's speedup model estimates the speedup of a job using its average parallelism, A, and its variance in parallelism, V. The average parallelism is the average parallelism of the job throughout its execution and the variance in parallelism is the change of parallelism of the job over time. The variance in parallelism is defined as $V = \sigma(A-1)^2$, where σ is the coefficient of variance in parallelism. The speedup formula for low parallelism variance, $\sigma < 1$, is given as

$$S_n = \begin{cases} \dfrac{An}{A + \sigma(n-1)/2}, & 1 \le n \le A \\ \dfrac{An}{\sigma(A-1/2) + n(1-\sigma/2)}, & A \le n \le 2A-1 \\ A, & n \ge 2A-1. \end{cases}$$

For high parallelism variance, $\sigma \ge 1$, it is as follows:

$$S_n = \begin{cases} \dfrac{nA(\sigma+1)}{A + A\sigma - \sigma + n\sigma}, & 1 \le n \le A + A\sigma - \sigma \\ A, & n \ge A + A\sigma - \sigma \end{cases}$$

where n is the number of processors. Using this speedup estimation, a resource can estimate the execution time of a job and schedule submitted jobs over multiple processors. There are several scheduling strategies in [26]. In our simulations, we use a simple scheduling strategy which allocates a number of processors equal to the average parallelism of the job, A. If A processors are not available at time of the job request, the resource postpones the execution of this job until A processors become available.

The processing characteristics of jobs are determined by three parameters in our simulations: job instruction count in million instructions (MI), average parallelism and variance in parallelism. Also, resources are characterized with the number of processors and the processing speed of each processor in terms of million instructions per second.

We choose the job instruction count to be distributed uniformly between 100 and 3,000 MI, average parallelism distribution between 1 and 20 and parallelism variance distribution between 0 and 2. Resources are characterized with the number of processors and the processing speed of each processor in terms of million instructions per second. In simulations, each computational resource has 5,000 processors and each processor has a processing power of 20,000 million instructions per second (MIPS).

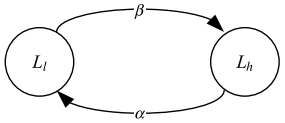

Fig. 5. State transition diagram of the MMPP traffic model.

8.3. Best-effort traffic model

The characteristics of the best-effort traffic which shares the network resources with the grid traffic have an important effect on the performance of the path selection algorithm. If the best-effort traffic is showing too little change over time, a path switching algorithm will not be necessary since the loss rates of alternative paths rarely change. On the other hand, if the distribution of the best-effort traffic over the network is fluctuating, a path switching algorithm performs much better than a static routing algorithm.

We use an MMPP traffic model to emulate the best-effort traffic because the traffic load in communication networks is bursty. In the simulations, each edge router keeps an average of three flows at the same time to different edge routers and each flow has an average holding time of 120 s. Bursts belonging to these flows are generated according to an MMPP distribution. One of the states of the MMPP distribution is the high load state and the other one is the low load state, as shown in Fig. 5.

The burst arrival rates at the states of an MMPP flow are determined according to a burstiness factor $\gamma \le 1$. The traffic load is $L_h = L_{Av}/\gamma$ in the high load state and $L_l = L_{Av}\gamma$ in the low load state. We determine the average load per flow, L_{Av}, to satisfy a desired offered load level on each link using the following formula:

$$L_{Av} = \frac{L * N_{links}}{E * F * N_{hops}}$$

where L is the desired average load level per link and N_{links} is the number of links in the network. E denotes the number of edge routers and F is the average number of best-effort flows originating from an edge router at a time; we select $F = 3$ in our simulations. N_{hops} is the average number of hops that best-effort bursts travel.

The transition rates of the MMPP distribution, α and β, are determined such that the average load per flow, L_{Av}, is satisfied. First, the state probabilities are found by solving these two equations:

$$L_l p_l + L_h p_h = L_{av}$$
$$p_l + p_h = 1.$$

Next, the transition rates can be found by selecting an appropriate value for one of the transition rates, α and β, and computing the other one using the formula

$$p_l = \frac{\alpha}{\alpha + \beta}.$$

Using this model, it is possible to experiment with different burstiness levels by changing the value of γ. The traffic generated by each flow is static for $\gamma = 1.0$. When γ is small, the generated traffic becomes more bursty.

8.4. Compared algorithms

To evaluate the performance of joint resource and path selection, we need to compare our proposed mechanism against other possible algorithms which perform resource and path selection separately. The most reasonable resource selection strategy is choosing the resource which offers minimum completion time if resource selection is considered independent of the path selection since the single metric we want to optimize is the completion time of a job. We call this method MCR. To choose the path to the selected resource, we use two path switching algorithms given in [16] for comparison:

- Weighted Link Congestion Strategy (WLCS): This path switching strategy computes the successful transmission probability of a path using the loss reports of each core router and weights this probability using the hop count of the path when selecting a path.
- Weighted Bottleneck Link Utilization Strategy (WBLU): This algorithm uses the utilization value of the most congested link along a path weighted by the hop length and selects the path accordingly.

In addition to these two path switching methods from the literature, we use shortest path routing (SP) in order to quantify the advantages of path switching.

The proposed joint resource and path selection algorithm presented in Section 5 is denoted as JR, and its extension with the adaptive offset mechanism given in Section 6 is denoted as JR-AO. To clearly evaluate the effect of adaptive offset mechanism, we compare JR-AO with JR-NO and JR-FO, corresponding to JR with no extra offset and JR with fixed offset, respectively.

8.5. Stationary best-effort traffic scenario

Dynamic path switching and adaptive offset schemes are expected to give better results under dynamic traffic loads because of their ability to react to changes in the network. However, the proposed algorithms are first compared for a stationary best-effort traffic load. In this case, "stationary" means that the average best-effort traffic load per link does not change over time. However, since each flow generates MMPP traffic, the traffic distribution is still bursty for $\gamma < 1$.

Simulations under stationary best-effort traffic are performed for different values of the best-effort traffic load and burstiness factors.

8.5.1. Effect of increasing best-effort traffic load

For $\gamma = 1.0$ and a grid load of 0.1 Erlangs, the graphs of average completion time and average offset for increasing best-effort traffic load are given in Figs. 6 and 7, respectively.

In terms of the completion time, it can be seen that JR algorithms perform better than MCR-WBLU and MCR-WLC, which perform resource selection and path selection separately, and also better than MCR-SP, which uses shortest path routing. JR-AO performs better than JR-FO, which is also better than JR-NO. Since JR-AO determines the offset value adaptively, it outperforms JR-FO, which applies a static offset to every burst in the network. JR-AO reduces the completion time up to 5% in comparison to JR-FO and 10% in comparison to MCR-WBLU. All of these algorithms show similar performance for low best-effort traffic load levels but their performance differences become more visible for higher best-effort traffic loads. For that reason, it can be deduced that the resource and path selection algorithm is not crucial for low loads.

Fig. 7 shows the average offset for job and result bursts for different best-effort traffic loads generated by JR-AO. As it can be observed from this graph, the average offset value applied to the job bursts are larger than the average offset of result bursts because the retransmission cost of job bursts is larger than the retransmission cost of result bursts. We use the average of the offset values generated by JR-AO for different best-effort loads as the fixed offset value in JR-FO. The fixed offset for JR-FO is chosen independently of the best-effort load in order to compare the proposed methods against a static priority scheme for grid bursts. The fixed offset value is the same for both job and result bursts. For that reason, the average fixed offset value is generally larger than the offset value of result bursts and smaller than the offset value of job bursts.

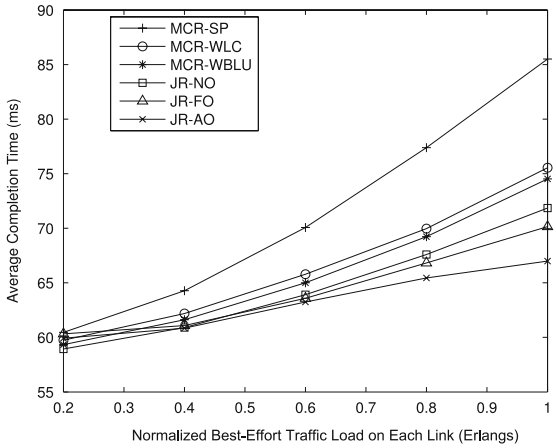

Fig. 6. Graph of average completion time versus offered best-effort traffic load for $\gamma = 1.0$.

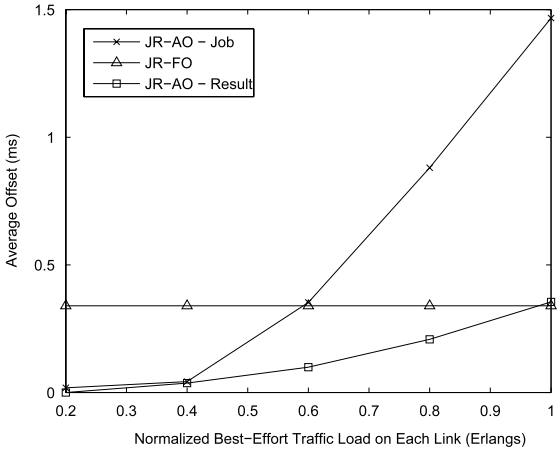

Fig. 7. Graph of average extra offset versus offered best-effort traffic load for $\gamma = 1.0$.

8.5.2. Effect of increasing burstiness

As the burstiness of the best-effort traffic load increases, the estimation of loss rates becomes more difficult. Several simulations with different burstiness factors are performed without changing the best-effort traffic load to evaluate the effect of the burstiness factor, γ, on the performance of the compared algorithms. Figs. 8 and 9 show the average completion time and average offset plots, respectively, for different burstiness levels when the best-effort traffic load per link is 0.4 Erlangs and the grid traffic load per link is 0.1 Erlangs.

In terms of the completion time, JR-AO performs best for all burstiness levels. As the burstiness increases, the average completion times for all algorithms increase but the performance of MCR-WBLU gets relatively better. The reason for this behavior is that MCR-WBLU uses the load level of the most congested link over a path for path selection. As the burstiness increases, the load differences between individual links become more significant, so using only the most congested link in path selection starts to perform better.

From Fig. 9, it can be observed that the average offset value increases as the burstiness of the best-effort traffic load increases, i.e., γ decreases. JR-AO applies higher offsets in the high load state of the MMPP distributed best-effort traffic so the average offset increases as the burstiness increases. Similar to Fig. 7, the average offset applied to job bursts is higher than the average offset applied to result bursts for all burstiness levels.

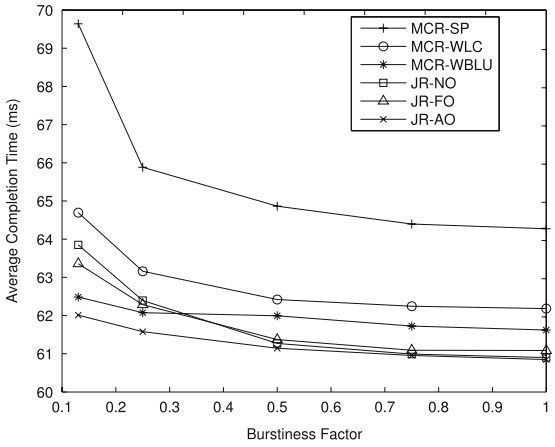

Fig. 8. Graph of average completion time versus burstiness factor γ.

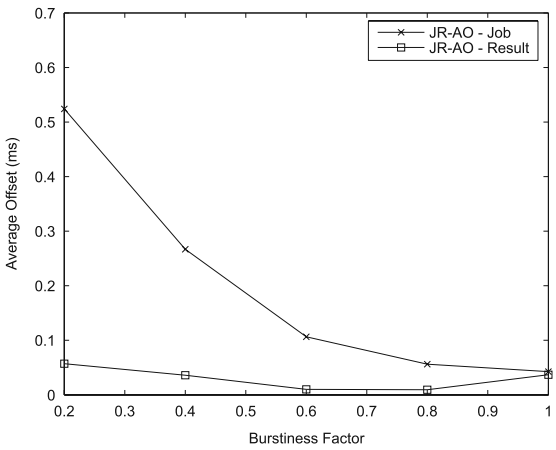

Fig. 9. Graph of average extra offset versus burstiness factor γ.

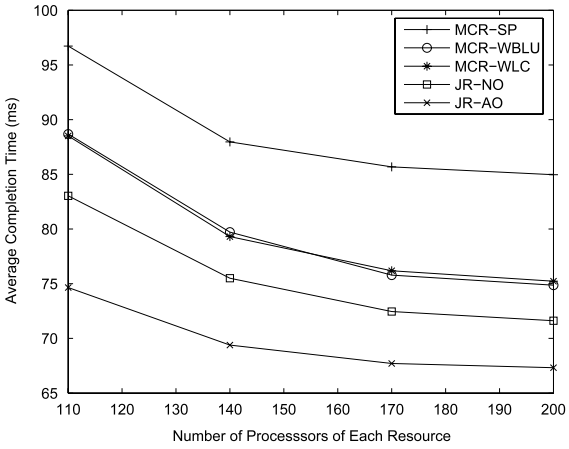

Fig. 10. Graph of average completion time versus number of processors for each resource.

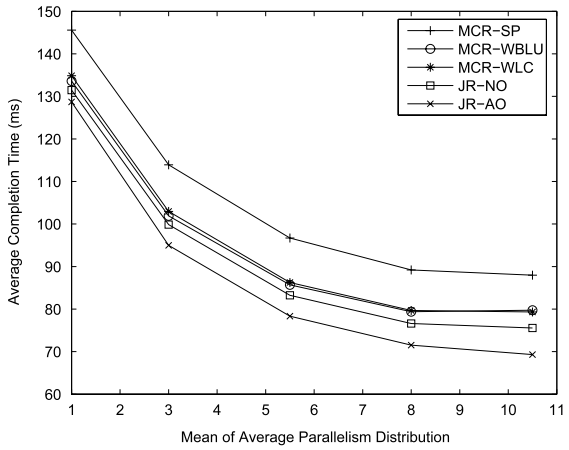

Fig. 11. Graph of average completion time versus mean of average parallelism.

8.5.3. *Effect of computational parameters*

In addition to parameters related to the network infrastructure, it is insightful to investigate the effect of change of grid job parameters on the average job completion time. Obviously, these parameters directly affect the completion times of jobs by changing the processing delays, but the network load levels are also affected by the resource and job parameters.

Fig. 10 shows the change of completion time for different numbers of processors for each computational resource obtained when the normalized best-effort traffic load is 0.8 and $\gamma = 1$. Since the jobs are queued at computational resources, the decrease in the number of processors results in increased completion times.

Fig. 11 shows the change of average completion time for increasing mean of average parallelism distribution, A. In this case, the maximum value of the uniform distribution is increased from 1 to 20; that is, the mean of the distribution function is increased from 1 to 10.5. Reduction of the parallelism increases the execution times of grid jobs because using multiple processors for a single job reduces the completion time.

In addition to affecting the completion time by changing the processing delays, a change in the job and resource parameters also has an impact on the network parameters. This impact is induced by the change in resource selection behavior of consumers when different grid parameters are used. For example, Fig. 12 shows the selection ratio of different resources by Consumer 1 as the number of processors at each resource increases. As the number of processors increases, the frequency of choosing Resource 1 increases up to a point after which Consumer 1 sends all of its jobs to Resource 1. The reason for this behavior is related to the distances of the resources from Consumer 1. As can be seen from Fig. 4, Resource 1 is the nearest resource to Consumer 1 and Resources 2 and 3 are far away. The completion times offered by resources are almost equal if there is no congestion at the resources and consumers tend to select the nearest resource to send their jobs because shortest paths generally have lower expected loss probabilities. However, some resources start to become more congested than others when there are fewer processors available and, consequently, consumers start to choose further away resources instead of congested nearby resources.

The impact of this change on the adaptive offset mechanism is shown in Fig. 13, where the average adaptive offset values of job and result bursts are plotted against the number of processors at each resource. As the number of processors decreases, the adaptive extra offset time increases because consumers send their bursts over longer paths and those paths have larger expected loss probability.

8.5.4. *Effect on the best-effort traffic load*

We have shown that joint resource-path selection and adaptive offset determination reduces grid job completion times. On the other hand, it is important to analyze the effect of these mechanisms on the blocking probability of the best-effort traffic in order to evaluate the drawbacks of this improvement.

Fig. 14 shows the burst loss probability of the best-effort traffic for a normalized background load of 0.8 with increasing grid traffic load. In this case, JR-NO achieves the lowest low priority burst loss probability followed by JR-FO, MCR-WLC and JR-AO, respectively. Although JR-AO shows the best performance in reducing the

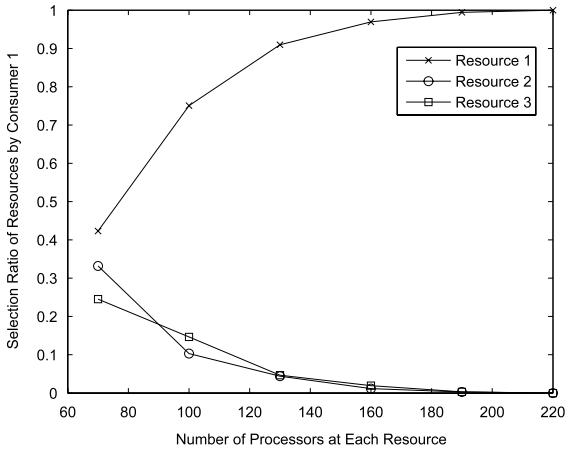

Fig. 12. Graph of ratio of selected resources by consumer 1 for different numbers of processors per resource.

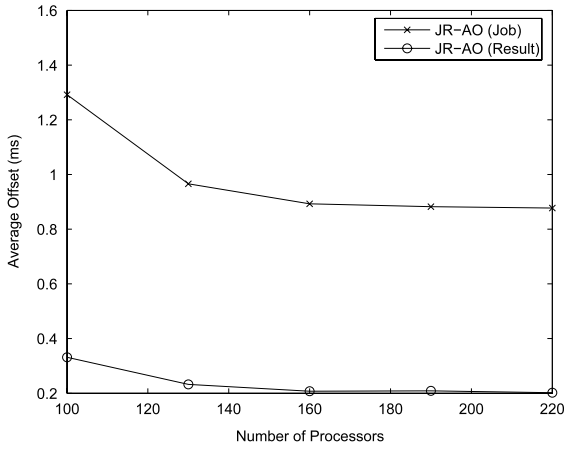

Fig. 13. Graph of average offset time versus processor count.

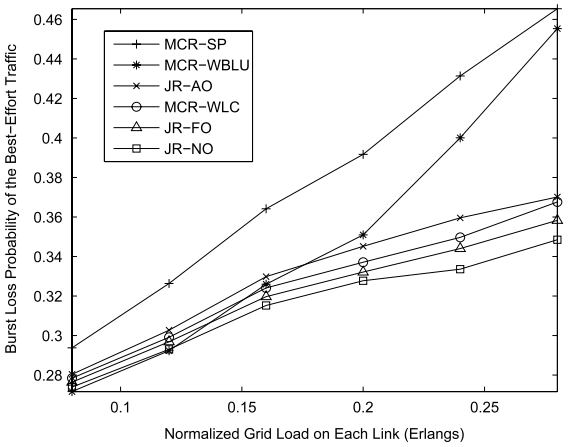

Fig. 14. Loss probability of the best-effort traffic as a function of the offered grid traffic load.

completion times, it increases the best-effort loss probability more because applying an offset to grid bursts increases the loss probability of the best-effort traffic. JR-FO has lower best-effort burst loss probability compared with JR-AO since JR-AO applies larger

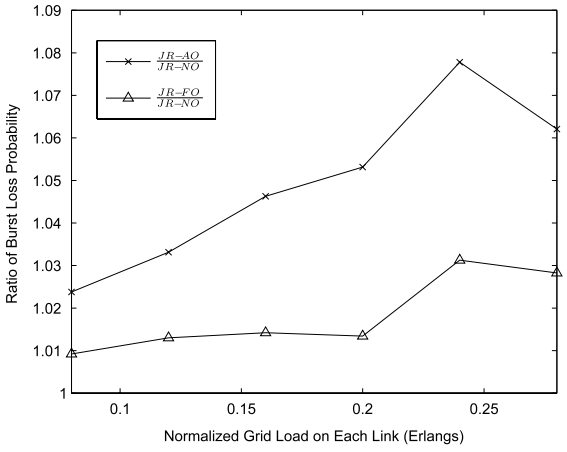

Fig. 15. Ratio of best-effort traffic loss probability for JR-NO and JR-AO in comparison to JR-NO.

offsets on average, as can be observed from Fig. 7. The increase in the loss probabilities for the best-effort traffic of JR-AO and JR-FO with respect to JR-NO are plotted in Fig. 15. It is observed that the loss probability of the best-effort traffic increases by less than 10% with JR-AO compared with JR-NO. On the other hand, applying only joint resource and path selection (JR-NO) does not increase the loss probability of the best-effort traffic because it tries to reduce the overall loss probability in the network by switching between resources and paths. Since there is no offset difference between best-effort and grid traffic in this mechanism, the loss probabilities of both types of traffic are same.

8.6. Non-stationary best-effort traffic scenario

In reality, the average traffic load in a network does not remain constant over time. The advantage of an adaptive congestion avoidance scheme is more significant in such a dynamic traffic load scenario because a fixed scheme cannot react to the fluctuations in the network traffic appropriately.

In this section, performances of the JR-AO, JR-FO and JR-NO are examined when the average best-effort traffic load is non-stationary. First, the reactions of the algorithms to a sudden increase in the best-effort traffic load are investigated. Later, the results for the case of a sudden decrease in the load are presented.

8.6.1. Sudden increase in best-effort traffic load

In Fig. 16, several performance metrics in the case of a sudden increase of the best-effort traffic load are plotted. In this scenario, between 400 s and 600 s the average best-effort traffic load is 0.8 Erlangs, and it is increased to 4 Erlangs at 600 s. The load is kept at that level until 800 s and, after that time, it is again reduced to 0.8 Erlangs. The first subplot shows the average offset value generated by JR-AO and the fixed offset value of JR-FO. The average offset generated by JR-AO in the low loss region is selected as the fixed offset value for JR-FO. The second subplot shows the change of average loss rate over time for JR-AO, JR-NO and JR-FO. The evolution of the average completion time is shown in the third subplot.

It can be observed from Fig. 16 that JR-AO reacts to the increase in the best-effort traffic load by increasing the offset values for grid bursts, and the benefit of this reaction can be observed in the loss rate and completion time graphs. There is a degradation in both metrics for all of the algorithms in the high load region, but the disadvantage of JR-AO is less than the other algorithms. The average completion time is reduced 20% by JR-AO in the high load region in comparison to JR-FO and 60% in comparison to JR-NO.

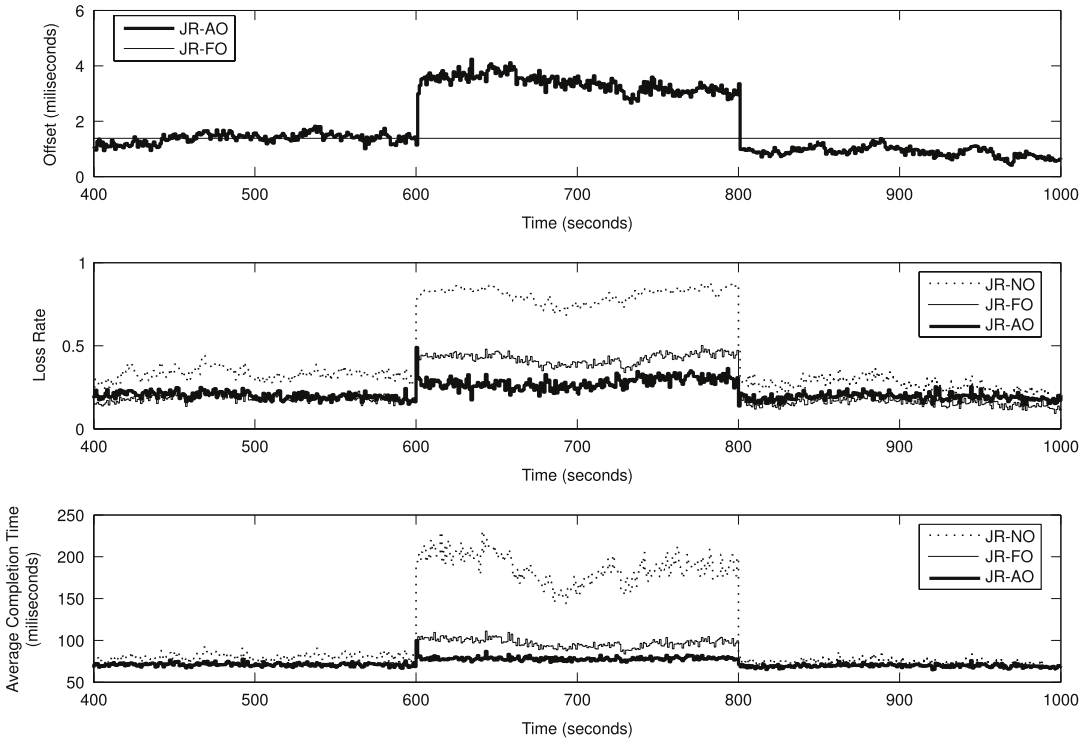

Fig. 16. Graph of change in average extra offset, loss rate and average completion time for a sudden increase in the best-effort traffic load for $\gamma = 1$.

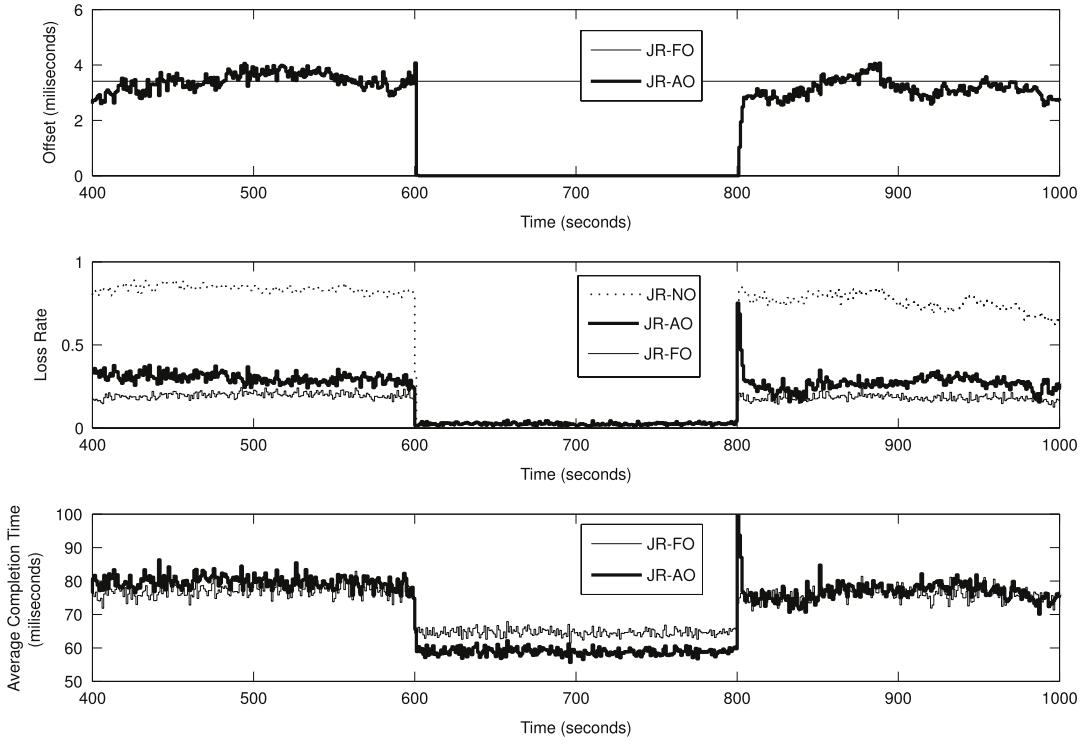

Fig. 17. Graph of change in average extra offset, loss rate and average completion time for a sudden reduction in the best-effort traffic load for $\gamma = 1$.

8.6.2. Sudden decrease in best-effort traffic load

A similar non-stationary scenario is the case of a sudden decrease in the best-effort traffic load level. In this scenario, the load level is kept at 4 Erlangs between 400 s and 600 s. After that the best-effort traffic load is completely removed until 800 s. Later, it is increased to 4 Erlangs again. In this case, the fixed offset value is selected according to a high load situation.

From Fig. 17, it can be seen that applying fixed extra offset increases completion time when the best-effort traffic load is zero. From the loss rate graph, it can be seen that JR-FO achieves better loss rates than JR-AO. However, JR-FO has no or little advantage over JR-AO in terms of completion time in the high loss region.

In the low loss region, the reduction obtained by JR-AO is approximately 6 ms, which is 8%. This amount is nearly the twice

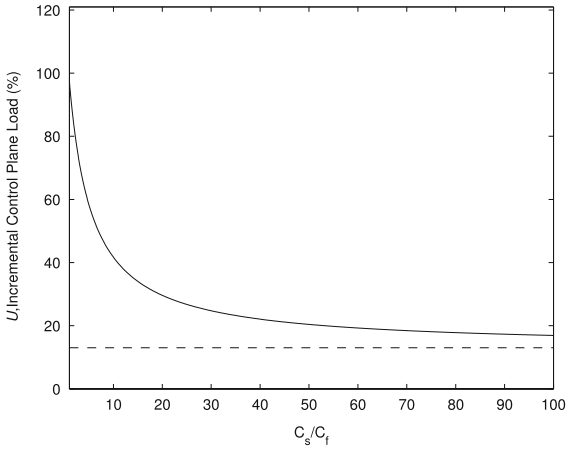

Fig. 18. Incremental load necessary for implementing the proposed mechanisms versus the ratio of the processing cost for scheduling and switch configuration to the processing cost for control packet forwarding.

the extra fixed offset, which is unnecessary in the low load region. The reason of this doubling effect is that the extra offset is applied to both of job and result bursts and the completion time of a grid job includes the extra offset of a job and result burst.

8.7. Control plane load

We algebraically formulated the control plane load created by the proposed mechanism in Section 7 and we evaluate those formulations for our simulated topology here. In the topology we simulated, $I = 3$, $N_{sp} = 4.86$, $N_{dp} = 5.50$ and $N_{mult} = 5.86$. The additional control plane load generated by the proposed mechanisms, U, as given in (15), is plotted in Fig. 18 as a function of C_s/C_f. For $C_s/C_f = 10$, our mechanisms add an extra control plane load of 41%. If $C_s \gg C_f$, the extra load converges to 13%, which is equal to the ratio $\frac{N_{dp}}{N_{sp}}$, which is also depicted in the figure.

9. Conclusions

As grid computing is expected to be used for short-lived consumer applications, a need for revisiting the grid architectural models emerged since current grid practices are focused on long-lasting and computationally intensive jobs. The OBS grid architecture is a dynamic, low latency grid model suitable for interactive applications with short-lived jobs. In this paper, we addressed the network and resource scheduling problems associated with OBS grids.

First, we formulated a joint resource and network provisioning method which reduces contention in the network by load balancing. The consumer selects a resource and a path to that resource in order to minimize job completion times. Second, we proposed an adaptive offset mechanism which increases the priority of grid bursts over the best-effort bursts by using an extra offset. Although applying an extra offset increases the delay in burst transmission, it reduces the burst losses, which in turn reduces the job completion times.

Simulation results show that proposed methods are successful in reducing the average completion time of grid jobs compared with the other path selection algorithms from the literature. The improvement is especially significant when the best-effort traffic shows short-term or long-term irregularities because the proposed algorithms can monitor changes in the network conditions and adapt their decisions. Also, we analyzed the effect of several grid parameters on the OBS grid performance. Results show that grid resource parameters affect the resource selection decision, especially when the grid resources start to congest.

These algorithms bring limited additional load on the control plane of the OBS network which is also analyzed mathematically. Negative effects of the proposed algorithms on the best-effort traffic are also investigated with simulations. Using only joint resource and path selection reduces the loss probability of the best-effort traffic as well as the grid traffic because it balances the load in the network, which is beneficial for both types of traffic. If the adaptive extra offset mechanism is used, the best-effort loss probability is increased but the increase is not significant.

Acknowledgements

This work is supported in part by the Science and Research Council of Turkey (Tübitak) under project EEEAG-104E047.

References

[1] The large hadron collider, http://lhc.web.cern.ch/lhc/.
[2] I. Foster, C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1998.
[3] R. Nejabati, et al., Grid optical burst switched networks, GOBS, Technical memo, Open Grid Forum, 2007.
[4] D. Simeonidou, R. Nejabati, et al., Optical network infrastructure for grid, Technical memo, Global Grid Forum, 2004.
[5] X. Yu, X. Liu, C. Qiao, T. Wang, Performance comparison of optical circuit and burst switching for distributed computing applications, in: Proceeding of Optical Fiber Communication/National Fiber Optic Engineers Conference, 2008, pp. 1–3.
[6] F. Palmieri, Network-aware scheduling for real-time execution support in data-intensive optical grids, Future Generation Computer Systems 25 (7) (2009) 794–803.
[7] T. Stevens, M.D. Leenheer, C. Develder, B. Dhoedt, K. Christodoulopoulos, P. Kokkinos, E. Varvarigos, Multi-cost job routing and scheduling in grid networks, Future Generation Computer Systems 25 (8) (2009) 912–925.
[8] M. Koseoglu, E. Karasan, Joint path and resource selection for OBS grids with adaptive offset based QoS mechanism, in: GridNets'07: Proceedings of the First International Conference on Networks for Grid Applications, 2007, pp. 1–8.
[9] D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, A. Tzanakaki, M. O'Mahony, Dynamic optical network architectures and technologies for existing and emerging grid services, Journal of Lightwave Technology 23 (10) (2005) 3347–3357.
[10] L. Berger, Generalized multi-protocol label switching (GMPLS) signaling functional description, RFC 3471, 2003.
[11] C. Qiao, Optical burst switching (OBS)—A new paradigm for an optical internet, Journal of High Speed Networks 8 (1) (1999) 69–84.
[12] M. De Leenheer, E. Van Breusegem, R. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, D. Simeonidou, M. Mahoney, R. Nejabati, T. Tzanakaki, I. Tomkos, An OBS-based grid architecture, in: Global Telecommunications Conference Workshops, 2004, pp. 390–394.
[13] M. De Leenheer, P. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, M.J. O'Mahony, A view on enabling-consumer oriented grids through optical burst switching, IEEE Communications Magazine 44 (3) (2006) 124–131.
[14] R. Samanta, T. Funkhouser, K. Li, Parallel rendering with k-way replication, in: PVG'01: Proceedings of the IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics, 2001, pp. 75–84.
[15] G. Thodime, V. Vokkarane, J. Jue, Dynamic congestion-based load balanced routing in optical burst-switched networks, in: GLOBECOM'03. IEEE Global Telecommunications Conference, 2003, pp. 2628–2632.
[16] L. Yang, G. Rouskas, Adaptive path selection in OBS networks, Journal of Lightwave Technology 24 (8) (2006) 3002–3011.
[17] I. Katib, D. Medhi, Adaptive alternate routing in WDM networks and its performance tradeoffs in the presence of wavelength converters, Optical Switching and Networking 6 (3) (2009) 181–193.
[18] E. Karasan, E. Ayanoglu, Effects of wavelength routing and selection algorithms onwavelength conversion gain in WDM optical networks, IEEE/ACM Transactions on Networking 6 (2) (1998) 186–196.
[19] A. Barradas, M. Medeiros, Edge-node deployed routing strategies for load balancing in optical burst switched networks, ETRI Journal 31 (1) (2009) 31–41.
[20] B.-C. Kim, Y.-Z. Cho, J.-H. Lee, Y.-S. Choi, D. Montgomery, Performance of optical burst switching techniques in multi-hop networks, in: Global Telecommunications Conference, GLOBECOM, vol. 3, 2002, pp. 2772–2776.
[21] M. Yoo, C. Qiao, A new optical burst switching protocol for supporting quality of service, in: Proceedings of SPIE, 1998, pp. 396–405.
[22] K. Dolzer, C. Gauger, J. Späth, S. Bodamer, Evaluation of reservation mechanisms for optical burst switching, AEU International Journal of Electronics and Communications 55 (1) (2001) 18–26.

[23] D.L. Jagerman, Some properties of the Erlang loss function, Bell System Technical Journal 53 (3) (1974) 525–551.
[24] A. Iosup, C. Dumitrescu, D. Epema, H. Li, L. Wolters, How are real grids used? The analysis of four grid traces and its implications, in: IEEE/ACM International Workshop on Grid Computing, 2006, pp. 262–269.
[25] A model for moldable supercomputer jobs, in: IPDPS'01: Proceedings of the 15th International Parallel and Distributed Processing Symposium, 2001.
[26] A.B. Downey, A parallel workload model and its implications for processor allocation, Cluster Computing 1 (1) (1998) 133–145.

Mehmet Koseoglu received his B.Sc. and M.Sc. degrees from the Electrical and Electronics Engineering Department, Bilkent University, Ankara, Turkey in 2004 and 2007, respectively. From to 2004 and 2006, he worked as a software engineer for Aselsan Inc, Ankara, Turkey. He is currently a Ph.D. candidate in the Electrical and Electronics Engineering Department of Bilkent University. His research interests include optical grid networks and wireless multiple access protocols.

Ezhan Karasan received his B.S. degree from Middle East Technical University, Ankara, Turkey, M.S. degree from Bilkent University, Ankara, Turkey, and Ph.D. degree from Rutgers University, Piscataway, New Jersey, USA, all in electrical engineering, in 1987, 1990, and 1995, respectively. In 1995–1996, he was a post-doctorate researcher at Bell Labs, Holmdel, New Jersey, USA. From 1996 to 1998, he was a Senior Technical Staff Member in the Lightwave Networks Research Department at AT&T Labs-Research, Red Bank, New Jersey, USA. He has been with the Department of Electrical and Electronics Engineering at Bilkent University since 1998, where he is currently an associate professor. Dr. Karasan is a member of the Editorial Board of the journal Optical Switching and Networking. He is the recipient of the 2004 Young Scientist Award from the Turkish Scientific and Technical Research Council (TUBITAK), 2005 Young Scientist Award from Mustafa Parlar Foundation and a Career Grant from TUBITAK in 2004. Dr. Karasan received a fellowship from the NATO Science Scholarship Program for overseas studies in 1991–94. Dr. Karasan has been participating in FP6-IST Network of Excellence (NoE) e-Photon/ONe+ and FP7-IST NoE BONE projects. His current research interests are in the application of optimization and performance analysis tools for the design, engineering and analysis of optical networks and wireless ad hoc/mesh/sensor networks.