A Dynamically Adaptive Hybrid Algorithm for Scheduling Lightpaths in Lambda-Grids

Neena R. Kaushik and Silvia M. Figueira
Department of Computer Engineering
Santa Clara University, Santa Clara, CA 95053-0566, USA
{nrkaushik, sfigueira}@scu.edu

Abstract

This paper focuses on the scheduling of multiple-wavelength lightpaths and outlines an algorithm to provide increased simultaneous allocations of lightpaths on the Lambda-Grid. Several approaches for lightpath scheduling have been proposed, and most of them involve some variation of concentrating or balancing wavelengths over the edge-disjoint paths between the source and destination specified in a request. We have used our LRSS, Lightpath Request Scheduling Simulator, to show that the efficiency of balancing- and concentrating-based algorithms depends on the characteristics of the lightpaths requested. Based on this assumption, we have developed a dynamically adaptive hybrid algorithm, which combines both balancing and concentrating, employing each of the approaches when appropriate. The efficiency of the new algorithm is assessed by comparing its performance with the performance of the two basic approaches. Our experiments show that the dynamically adaptive hybrid scheduling algorithm has a blocking probability that is always close to the lower one, for a representative set of topologies under a representative set of traffic conditions.

1. Introduction

Optical networks are being deployed today in state-of-the-art Lambda-Grid testbeds, such as the OMNInet [15], Teragrid [25], SURFnet [24], CAnet [1], and others worldwide. It has been shown that dynamically provisioned optical networks provide the sustainable bandwidth needed by data-intensive grid applications and also enables the usage of the network by on-demand and advance reservation [1], which is key for remote collaboration. For these reasons, it is clear that, in the coming years, optical networks will be the main component in grid platforms built for data-intensive computing. In fact, this tendency is shown by recent projects such as the OptIPuter [22].

The importance of advance-reservation for grid computing, in particular for grid collaborations, has been introduced by the Globus group in the definition of GARA [3] and then in [4]. Recently, it has been discussed in [5], which focuses on data-intensive collaboration, and in [11], which aims at scheduling data placement activities. Advance reservation of lightpaths in Lambda-Grids will provide bandwidth guarantees for the specified time-slot [2].

Network reservation, either on demand or in advance, requires scheduling. When dealing with optical networks, this scheduling is done in terms of time-slots/ lightpaths, i.e., end-to-end lightpaths are reserved for a specific time slot. The scheduling of advance-reservation requests in grid environments has been discussed in [1], and advance reservations for packet-switched networks have been discussed in [19, 20].

The advance-reservation scheduling of lightpaths in optical networks consists basically of routing and wavelength assigning, an NP-complete problem, which has been extensively studied for on-demand reservations. This problem has been studied both with and without wavelength converters, and lower and upper bounds have been found for the number of wavelengths and number of wavelength converters needed for a given upper bound on the traffic [17]. In fact, in [18], the lower and upper bounds on the traffic have been defined based on the traffic pattern using integer linear programming. These bounds relate to the lower bound on blocking probability.

Lightpath scheduling in an optical network without wavelength converters consists of two main steps: selecting the route and selecting the wavelength within the route. Given the two main steps, generally, any scheduling approach will either (1) select a route and then pick a wavelength within the route or (2) select a wavelength and then pick a route within which that wavelength is available. Different selection processes may be employed in each approach, leading to different levels of optimizations. When scheduling multiple-wavelength lightpaths, the first approach can be extended to (3) select a set of routes and then pick wavelengths within the routes, while the second approach can be extended to (4) select a set of wavelengths and then pick routes within which those wavelengths are available. In approach number 3, the wavelengths will be spread within the set of routes, whereas, in approach number 4, the wavelengths will be packed over the set of routes. Packing and spreading are, in fact, basic techniques used for lightpath scheduling, on top of which several approaches have been built (see Section 2). We limit our studies to networks which do not have wavelength converters. This implies that, for a lightpath allocation, the same wavelength of light should be used from source to destination.

In [2], we defined and compared two algorithms, Wavelength-Balancing and Wavelength-Concentrating, which systematically schedule multiple-wavelength lightpaths by packing and spreading the wavelengths, respectively. These algorithms are explained in Section 3. In this paper, we analyze and compare the blocking probability obtained when wavelengths are balanced and concentrated in different scenarios and larger networks. The blocking probability was obtained with LRSS, our Lightpath Request Scheduling Simulator. Our experiments led to the conclusion that each approach behaves better than the other under special conditions, and that the length of the shortest path between the source and destination in the requests definitely affects the behavior of the algorithms. Based on this assumption, we have developed a dynamically adaptive hybrid algorithm, which either balances or concentrates wavelengths depending on the length of the shortest path between the source and destination in each request.

Note that our dynamically adaptive hybrid algorithm is based on the systematic and basic balancing and concentrating algorithms. Enhancing the Hybrid algorithm to include optimization techniques in the selection of the route and/or wavelength is possible, but further experiments are required to assess the benefits obtained by each extension.

This paper is organized as follows. Section 2 presents related work. Section 3 discusses the extreme behavior of the balancing and concentrating algorithms and shows that each of them has a lower blocking probability than the other depending upon the characteristics of the lightpath requests. Section 4 presents the dynamically adaptive hybrid scheduling algorithm. Section 5 presents the simulation results, which compare the balancing, the concentrating, and the dynamically adaptive hybrid scheduling algorithms. Section 6 summarizes and concludes.

2. Related Work

A range of heuristics have been proposed for the RWA (Routing and Wavelength Assignment) problem. In the order of performance (from worst to best), the following approaches have been proposed: *basic*, *porder*, *color*, *lpcolor*, *least-loaded*, *aurpack*, *aurexhausted* [8]. When comparing the average blocking probability achieved by each approach, spread generally provides the worst allocations for wavelength assignment. Packing performs better, and the performance increases from using wavelengths in order (porder), to using the most used wavelength (pcolor), to using the routes with most used wavelengths (lpcolor), to choosing the path which is left with the highest capacity after the assignment (least loaded), to using aurpack and aurexhaustive [8].

The authors in [8] define the various heuristic algorithms along with simulation results for a network in Finland, and show that the choice of heuristic influences the blocking probability of the network topology. The authors have also introduced a first-iteration policy to improve the blocking probability of the various algorithms by iteration. The iteration policy runs a simulation, then finds the cost and decides whether to accept or reject the connection along that path. It is important to note that the first iteration policy is computationally intensive. In [7], the authors define a method for path selection, which is based on exchanging information about critical links in the network and avoiding those links during wavelength assignment. It has been shown that this method reduces the blocking probability compared to a fixed-wavelength-assignment scheme. This method has an overhead imposed by the exchange of network link-state information. It has been shown in [6] that using the alternate path for wavelength assignment only when it is lightly loaded lowers the blocking probability of the network.

3. Balancing vs. Concentrating Wavelengths

This section presents the Wavelength-Balancing and Wavelength-Concentrating Algorithms and shows that each of them is the best choice under specific traffic and topology conditions.

3.1. Definitions

A. *Lightpath Request*: Each request is specified by a source-destination pair, a time-slot, and a number of wavelengths. It can be granted or rejected depending on the current availability for the specified time-slot. Note that granting a request means allocating one lightpath from source to destination for each wavelength requested.

B. *Lightpath Allocation*: A request that was granted for an end-to-end path (source-destination) and is specified in terms of number of wavelengths.

C. *Lightpath Rejection:* A lightpath request that was not granted for an end-to-end path (source-destination), specified in terms of number of wavelengths.

D. *Edge-Disjoint Path*: The *n*th edge-disjoint path between two nodes is a path that does not share any of its edges with the previous *n* - 1 edge-disjoint paths between those two nodes. The first edge-disjoint path is the shortest path, followed by the second shortest, and so on.

E. *End-to-End Path*: A path that starts from the specified source and ends at the specified destination. The minimum number of links in an end-to-end path is one.

F. *Blocking probability*: The ratio of total rejections to the total requests. It is given by

$$P_B = \rho / \alpha,$$

where ρ is the total number of rejections, and α is the total number of requests.

3.2. The Algorithms

- *Wavelength-Concentrating Algorithm:* This algorithm finds all the edge-disjoint paths from the source to the destination. It then allocates all the wavelengths along the first edge-disjoint path, then allocates all wavelengths along the second edge-disjoint path and so on, sequentially until all the requests are satisfied [2].
- *Wavelength-Balancing Algorithm:* This algorithm finds all the edge-disjoint paths from the source to the destination. It then allocates the first wavelength along the first edge-disjoint path, the first wavelength along the second edge-disjoint path, and so on, along the *n*th edge-disjoint path. It then allocates the second wavelength in the same order and so on, up to the *m*th wavelength until all the requests are satisfied [2].

3.3. LRSS

The blocking probabilities for various topologies have been found using LRSS, our Lightpath Request Scheduling Simulator. LRSS is a simulator written in C, which takes as input the network topology and a synthetic trace of requests for lightpaths, both in the form of ASCII text files. For the set of experiments in this Section, we used FONTS (Flexible Optical Network Traffic Simulator) [13] traces for the lightpath requests.

There are no real traces for advance-reservation Lambda-Grids available at the moment. In [9], it has been emphasized that, in the absence of real traces for high bandwidth networks, the traces can be generated by modeling the network behavior using stochastic processes, which is exactly how FONTS generate its synthetic traces. In our case, the FONTS traces were generated for a single source-destination pair, and the source-destination pairs were modified by LRSS according to the traffic characteristics in Table 1, to generate specific kinds of requests.

LRSS simulates scheduling algorithms for advance reservations. The algorithms currently simulated are: Wavelength-Balancing, Wavelength-Concentrating, and the dynamically adaptive hybrid scheduling algorithm which combines the balancing and concentrating schemes. An important part in these algorithms is finding the edge-disjoint paths between the given source-destination pair of a lightpath request. The edge-disjoint paths are produced by repeatedly applying the Dijkstra's shortest path algorithm to the graph produced by removing the edges, which were part of the shortest path produced in the previous step. The code for Dijkstra's shortest path appears in [21]. In using this approach, it is assumed that the capacity of each link is one unit. In [16], it has been explained that the *right* approach to find the edge-disjoint paths involves decrementing by one unit the capacity of the links in each newly-found edge-disjoint path. Since, in our case, the capacity of each edge is one unit, this approach reduces to ours because reducing a one-unit edge's capacity to zero unit is the same as eliminating the edge.

Table 1: Algorithms for generating different types of traffic.

8-hop Requests	*source = rand() % ring_total_nodes;* *if (8 > (ring_total_nodes - source))* *dest = 8 - (ring_total_nodes - source);* *else* *dest = source + 8;*
Uniform Requests	*source = rand() % ring_total_nodes;* *dest = rand()% ring_total_nodes;* *while (source == dest)* *dest = rand()% ring_total_nodes;*
1-hop Requests	*source = rand() % ring_total_nodes;* *if (source + 1 < ring_total_nodes - 1)* *dest = source + 1* *else* *dest = 0;*

3.4. Extreme Cases

The Wavelength-Balancing and Wavelength-Concentrating algorithms have been analyzed extensively for 4-node topologies and the OMNInet in [2]. The Wavelength-Balancing algorithm performs better when the number of two-hop source-destination requests were biased to 25, 50, and 75 percent compared to the number of one-hop source-destination requests. The Wavelength-Balancing algorithm also performed better when the diameter link of the OMNInet topology was used for satisfying the lightpath requests, but was not requested as an end-to-end lightpath. However, the Wavelength-Concentrating algorithm performed better when the requests were mainly for one-hop end-to-end lightpaths.

To illustrate the variation in performance according to the type of request, we show extreme cases, in which the Wavelength-Balancing and Wavelength-Concentrating algorithms performed better than each other. The simulation was performed for a 32-node-ring and a 32-node-ring-with-chords topologies, in which the capacity in each link is four wavelengths. The representative examples of kinds of requests which lead to better performance for the Wavelength-Balancing algorithm and for the Wavelength-Concentrating algorithm are shown in Tables 2 and 3 respectively.

Table 2: 32-node-ring topology: requests for 1-hop lightpaths.

Variable	Value
Advance reservation request arrival	Poisson distribution
Average number of lightpath request arrivals in a time-slot	60, 30, 20, 15, 12 and 10
Number of lightpath requests for each arrival	Uniform [1-4]
Time-slot length	1 hour
Source-destination pair distribution	Uniform traffic between 1-hop-apart nodes only

Table 3: 32-node-ring-with-chords topology: requests for 8-hop lightpaths.

Variable	Value
Advance reservation request arrival	Poisson distribution
Average number of lightpath request arrivals in a time-slot	60, 30, 20, 15, 12 and 10
Number of lightpath requests for each arrival	Constant = 1
Time-slot length	1 hour
Source-destination pair distribution	Uniform traffic between 8-hop-apart nodes only

Figures 1 and 2 show the blocking probability for both concentrating and balancing when the requests are as specified in Tables 2 and 3 respectively. Table 1 shows how the source-destination pairs were generated by LRSS for the extreme cases. The traffic is assumed to be unidirectional.

The graph in Figure 1 shows that, for a 32-node-ring topology, when the requests are for 1-hop lightpaths only, the Wavelength-Concentrating algorithm performs better. However, for a 32-node-ring-with-chords topology (see Figure 6), when the requests are for 8-hop lightpaths only, the Wavelength-Balancing algorithm performed better, as shown in Figure 2.

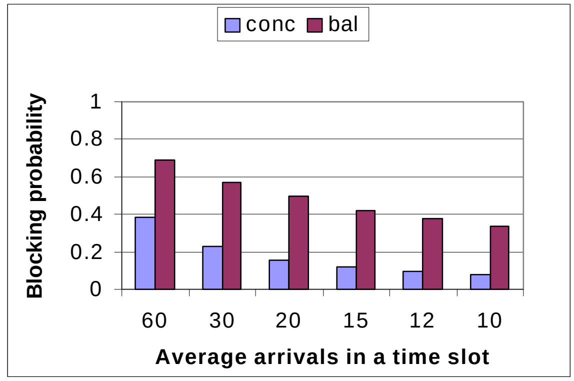

Figure 1: Requests for 1-hop lightpaths in a 32-node ring: uniform requests

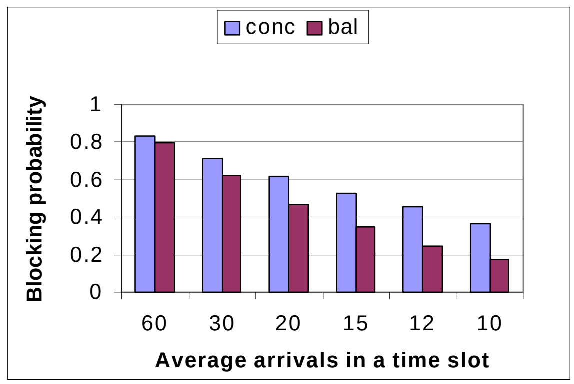

Figure 2: Requests for 8-hop lightpaths in a 32-node ring-with-chords: constant requests.

4. The Hybrid Algorithm

We have developed a novel dynamically adaptive hybrid scheduling algorithm, which is a combination of the balancing and concentrating schemes. This algorithm concentrates the allocations along the shortest paths or balances them along all the edge-disjoint paths depending on the length of the source-destination path in the lightpath request. It balances the allocations along the edge-disjoint paths which are shorter than a threshold and concentrates the allocations along the edge-disjoint paths which are longer than or equal to the same threshold.

The Hybrid algorithm, shown below, runs the Wavelength-Balancing algorithm in two passes. In the first pass, it avoids those edge-disjoint paths which have more than *x*-hops. This way it achieves its concentrating effect in the first pass, by allocating only along those routes which have less than *x*-hops. In the second pass, it runs the Wavelength-Balancing algorithm again and allocates the remaining requests along all edge-disjoint paths. Note that *x* represents the cutoff value between balancing and concentrating.

Hybrid Algorithm

begin
  firstpass = 1;
  ***while** (firstpass is not equal to 3)*
    ***for** i = 1 to number of wavelengths*
      ***for** j = 1 to number of edge-disjoint paths*
        ***if** first pass is equal to 1 and*
            edge-disjoint path has more than x-hops
          continue
        ***if** wavelength[i] is available for all segments*
            in the edge-disjoint path[j]
          Allocate wavelength[i] for all segments
            in edge-disjoint path[j]
          Increment number of allocated requests by 1
        ***if** all requests are satisfied*
          return the number of requests satisfied
      ***end** (for edge-disjoint paths loop)*
    ***end** (for number of wavelengths loop)*
    Increment firstpass by 1
  ***end** (while loop)*
  return the number of requests satisfied
end

The graphs in Figures 3 and 4 illustrate the improvement gained with the adaptability of the Hybrid algorithm. They show the blocking probability obtained by the different algorithms (Balancing, Concentrating, and Hybrid with various cutoff values) when dealing with the extreme cases shown in Tables 2 and 3. The topology used in the experiments in Figure 3 is a 32-node ring and, in Figure 4, is a 32-node ring-with-chords. The chords connect the nodes (0, *n*/2-1), (0, *n*/4 -1), and (0, 3*n*/4 -1) where *n* is the number of nodes in the ring, as shown in Figure 6.

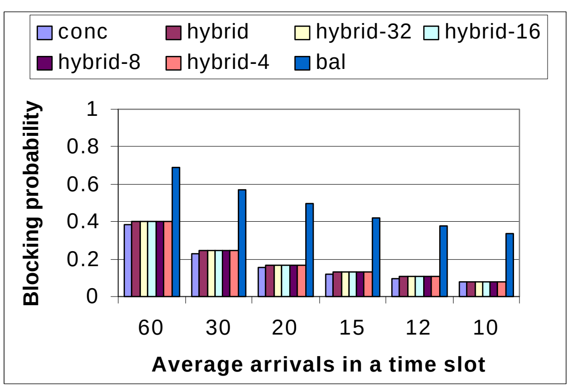

Figure 3: Requests for 1-hop lightpaths in a 32-node ring: uniform requests

Note that, in both cases, the blocking probability obtained by the Hybrid algorithm is comparable with the blocking probability obtained with the best algorithm for each case. In Figure 3, the requests are for 1-hop end-to-end paths, and the Hybrid algorithm performed comparably to the Wavelength-Concentrating algorithm, the best option in this case. In Figure 4, the requests are for 8-hop end-to-end paths, and the Hybrid algorithm performed comparably to the Wavelength-Balancing algorithm, the best option in this case. This example shows how the Hybrid algorithm is able to adapt to the requests and behave like the best option in each case.

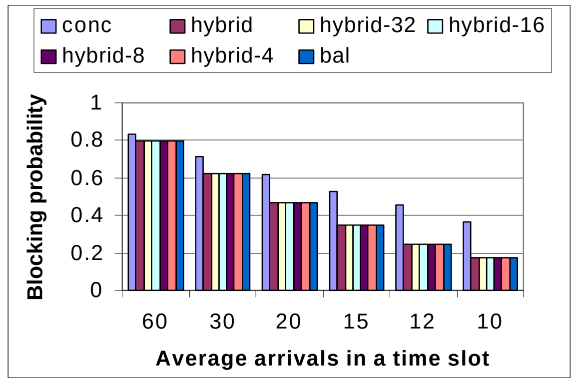

Figure 4: Requests for 8-hop lightpaths in a 32-node ring-with-chords: constant requests.

We have tried different cutoff values for the ring and ring-with-chords topologies, as shown in Figures 3 and 4, where:

- *Hybrid*: the algorithm balances along the edge-disjoint paths during the first pass, only if the number of hops in the edge-disjoint path is less than half of the total number of nodes.
- *Hybrid-n*: the algorithm balances along the edge-disjoint paths during the first pass, only if the number of hops in the edge-disjoint path is less than $N/2 + N/n$, where N is the total number of nodes. The *Hybrid-n* algorithm becomes the *Hybrid* algorithm when $n = \infty$.

In a ring topology every source-destination has two alternate paths. For a ring, the minimum length of the second-shortest edge-disjoint path is $N/2$. In a ring-with-chords topology, the minimum length of the second edge-disjoint path can be less than $N/2$. However we have used $N/2$, in our analysis for both ring and ring-with-chords topologies, as the minimum cutoff in order to study the effects of the Hybrid algorithm on both the ring and ring-with-chords topologies using the same cutoff parameters.

Note that, according to the graphs shown in Figures 3 and 4, in these two scenarios, varying the cutoff value does not seem to have a significant impact on the blocking probability.

5. Simulation Results

The results presented in this section show the blocking probabilities obtained from simulations using the LRSS with a 32-node ring and ring-with-chords topologies, as well as the topologies used in a soon-to-be production network, the National Lambda Rail, and in the SURFnet. We have experimented with the sequence of requests described in Table 4.

We consider ring topologies as representative topologies for our analysis because they are simple and are the basic building-blocks used to form more sophisticated topologies used in production networks, such as the OMNInet [15], SURFnet [24], and National Lambda Rail [14].

The subsections below present a representative set of the experiments performed using a constant distribution for the number of lightpaths in a request. A comprehensive set of experiments, including two other distributions for the number of lightpaths in a request, as shown in Table 4, can be found in [10].

Table 4: Requests for simple ring, ring-with-chords topologies National Lambda Rail and SURFnet.

Variable	Value
Advance reservation request arrival	Poisson distribution
Average number of lightpath request arrivals in a time slot	60, 30, 20, 15, 12 and 10
Number of lightpath requested	Constant = 1, Zipf’s distribution (exponent = 3, capacity = 4), Uniform Distribution [1 - 4]
Time-slot length	1 hour
Source-destination pair distribution	Uniform

Note that, the scenarios explored in the following subsections are not extreme, and the goal with these experiments is to show that the Hybrid algorithm is general and leads to a blocking probability, which is always comparable with the blocking probability obtained with the best algorithm in each of these cases.

Also note that we have used 1-hour time slots, which may not be the best choice in all the cases. Investigating the impact caused by varying the length of the time slot on the blocking probability, in different scenarios, is part of our future plans and is beyond the scope of this paper.

Another goal is to explore the impact of employing different cutoff values on the blocking probability obtained by the Hybrid algorithm. The results presented below show that varying the cutoff value did not impact the blocking probability significantly in the scenarios studied.

5.1. 32-Node Ring

In Figure 5, we show the results for constant requests in 32-node rings. Note that the difference in blocking probability obtained with the Wavelength-Balancing and Wavelength-Concentrating depends on the average arrivals in a time slot. Secondly, the Hybrid algorithm achieves the behavior of the algorithm which has the lower blocking probability.

Note that, in this scenario, according to the graph shown in Figure 5, varying the cutoff value does not have a significant impact on the blocking probability.

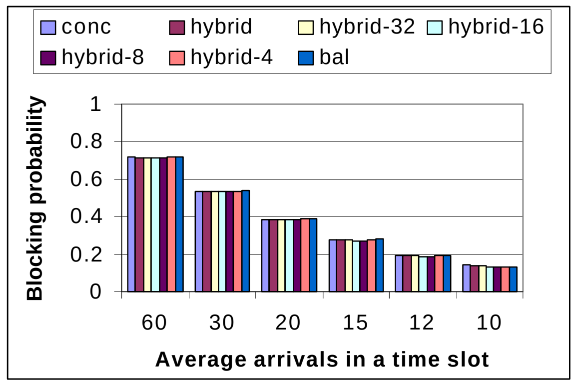

Figure 5: 32-Node Ring: Constant Requests

5.2. 32-Node Ring-With Chords

In Figure 7, we show the results for constant requests in 32-node rings-with-chords. The Wavelength-Balancing algorithm performs better than the Wavelength-Concentrating algorithm, and the Hybrid algorithm achieves a behavior similar to the Wavelength-Balancing algorithm, the best option in this case.

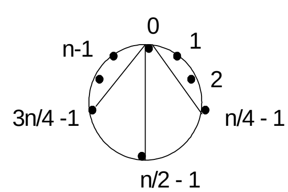

Figure 6: 32-Node Ring-With-Chords, n = 32.

Note that, as for the 32-ring topology, in this scenario, according to the graph shown in Figure 7, varying the cutoff value does not have a significant impact on the blocking probability.

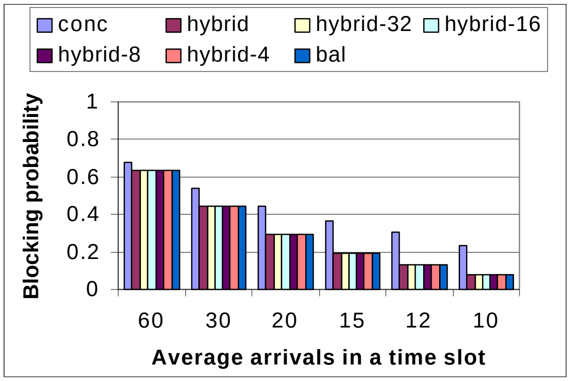

Figure 7: 32-Node Ring-With-Chords: Constant Requests

5.3. National Lambda Rail

The National Lambda Rail (as shown in Figure 8, which was obtained at [14]) is a soon-to-be-deployed optical network in the United States. In Figure 9, we present results for the NLR topology with constant requests. Note that different cutoff values provide slightly different blocking probabilities. Note also that, generally, the Hybrid algorithm performed as well as, or even better than the Balancing-Wavelength algorithm.

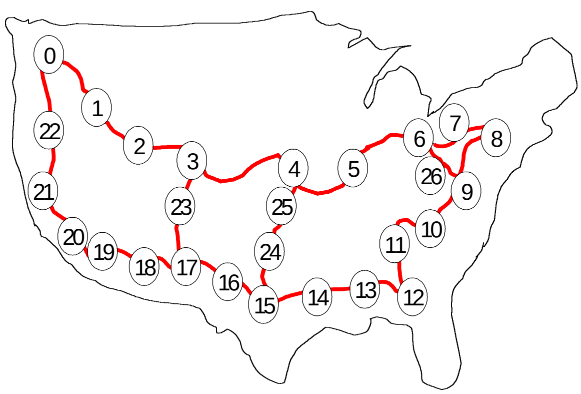

Figure 8: National Lambda Rail

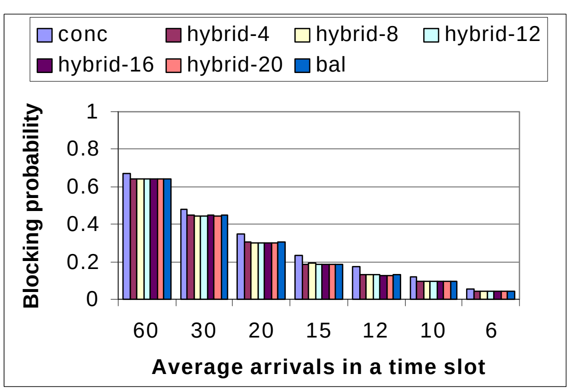

Figure 9: National Lambda Rail: Constant Requests

Since the NLR is not a ring, we have decided to experiment with absolute cutoff values, i.e., Hybrid-*x* indicates that the cutoff value used was *x*. In this scenario, according to the graph shown in Figure 9, varying the cutoff value does affect the blocking probability in some cases, but the impact is still not significant.

5.4. SURFnet

The SURFnet [24] is an initiative of the Dutch government to support research and educational activities. The core of SURFnet, a four-node ring as shown in Figure 10, is key in the cross-group communication. To reach nodes in the same access group, we assume no lambda scheduling is required. The lambda scheduling is required for reaching from one access group to another, which is done through the 4-node ring.

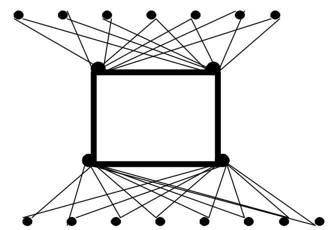

Figure 10: SURFnet

In Figure 11, we present results for the SURFnet with constant requests. Note that, generally, the Hybrid algorithm performed as well as, or even better than the best between Concentrating and Balancing, each of which was a better option in specific cases.

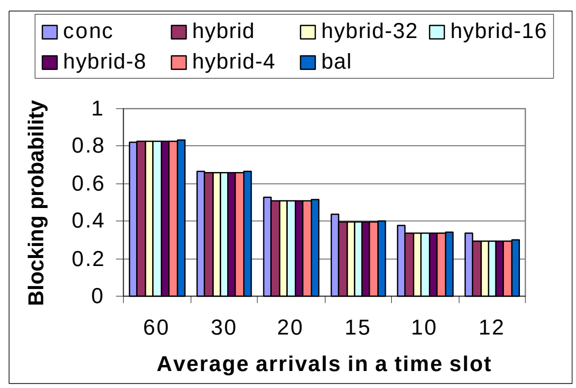

Figure 11: SURFnet: Constant Requests

In this scenario, as for the ring and ring-with-chord topologies, according to the graph shown in Figure 11, varying the cutoff value does not affect the blocking probability.

6. Conclusion

We have developed the LRSS (Lightpath Request Scheduling Simulator) to find the blocking probabilities of advance reservations of lightpath requests. Our simulation results show that the behavior of the scheduling algorithm depends on the lightpath request characteristics, and that both balancing and concentrating algorithms can achieve lower blocking probability, depending on the traffic characteristics.

Our novel dynamically adaptive hybrid scheduling algorithm, combines the concentrating and the balancing of wavelengths, and achieves the behavior of the algorithm which has the lower blocking probability, for ring, ring-with-chords, the National Lambda Rail, and the SURFnet topologies. As shown by our simulations, this algorithm has the potential to increase network utilization by enabling an increase in the number of simultaneous lightpath allocations on the Lambda-Grid.

Acknowledgement

We would like to thank Dr. Stephen Chiappari and Dr. Nirdosh Bhatnagar, from Santa Clara University, for their support. We would also like to thank Sumit Naiksatam, also from Santa Clara University, for providing the traces with advance-reservation requests used as input for LRSS.

References

[1] S. Figueira, et al, “DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks”, *CCGRID/GAN 2004, Workshop on Grid and Advanced Networks*, April 2004.

[2] S. Figueira, N. Kaushik, S. Naiksatam, S.A. Chiappari and N. Bhatnagar, “Advance Reservation of Lightpaths in Optical Network Based Grids”, *in ICST/IEEE GridNets*, San Jose, October 2004.

[3] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy, “A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation”, *International Workshop on Quality of Service*, 1999.

[4] I. Foster, A. Roy, and V. Sander, “A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation”, in *8th International Workshop on Quality of Service (IWQOS 2000)*, pp. 181-188, June 2000.

[5] I. Foster, J. Vockler, M. Wilde, and Y. Zhao, “The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration”, *Proceedings of the First CIDR - Biennial Conference on Innovative Data Systems Research*, January 2003.

[6] H.Harai, M. Murata and H. Miyahara, “Performance of Alternate Routing Methods in All-Optical Switching Networks”, *in Proceedings of the IEEE INFOCOM*, 1997, pp. 516-524, 1997.

[7] P. Ho and H.T. Mouftah, “A Novel Distributed Protocol for Path Selection in Dynamic Wavelength-Routed WDM Networks”, *Kluwer Photonic Network Communications*, Vol. 5, No. 1, pp. 23-32, January 2003.

[8] E. Hyytia and J.Virtamo, “Dynamic Routing and Wavelength Assignment Using First Policy Iteration”, *proceedings of the Fifth IEEE Symposium on Computers and Communications (ISCC 2000)*, p. 146, July 04-06, 2000.

[9] P. Kamath, K. Lan, J. Heidemann, J. Bannister and J. Touch, “Generation of High Bandwidth Network Traffic Traces”, *in 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Systems, (MASCOTS)*, Fort Worth, Texas, Oct 12-16, 2002.

[10] N. Kaushik and S. Figueira, “Analyzing a Dynamically Adaptive Hybrid Algorithm for Scheduling Lightpaths in Lambda-Grids,” *SCU COEN Tech Report*, March 2005.

[11] T. Kosar and M. Livny, “Stork: Making Data Placement a First Class Citizen in the Grid”, in *Proceedings of 24th IEEE International Conference on Distributed Computing Systems (ICDCS2004)*, March 2004.

[12] A. Mokhtar and M. Azizoglu, “Adaptive Wavelength Routing in All-Optical Networks”, *IEEE/ACM Transactions on Networking(TON)*, vol. 6, no. 2, pp 197-206, April 1998.

[13] S. Naiksatam, S. Figueira, S. Chiappari, and N. Bhatnagar, “Analyzing the Advance Reservation of Lightpaths in Lambda-Grids”, *CCGRID’05*, May 2005.

[14] National Lambda Rail: http://www.nlr.net/

[15] OMNInet: http://www.icair.org/omninet.

[16] M. Pioro and D. Medhi, “Routing, Flow, and Capacity Design in Communication and Computer Networks”, Morgan Kaufmann, 2004.

[17] R. Ramaswami and K.N. Sivarajan, “Routing and Wavelength Assignment in All-Optical Networks”, *IEEE/ ACM Transactions On Networking*, Vol 3 No 5, pp. 489-500, October 1995.

[18] R. Ramaswami and K.N. Sivarajan, “Optical Networks - A Practical Perspective”, Chapter 8, Morgan Kaufmann, 1998.

[19] O. Schelen and S. Pink, “Sharing Resources through Advance Reservation Agents”, *Proceedings of the IFIP International Workshop on Quality of Service*, May 1997.

[20] O. Schelen and S. Pink, “An Agent-based Architecture for Advance Reservation”, *Proceedings of the IEEE Conference on Computer Networks*, November 1997.

[21] S. Skiena and Miguel Revilla, “Programming Challenges: The Programming Contest Training Manual”, Springer-Verlag, New York, 2003.

[22] L. Smarr, et al, “The OptIputer”, *Special Issue: Blueprint for the Future of High Performance Networking, Communications of the ACM*, Vol. 46, No. 11, pp. 58-67, Nov. 2003.

[23] W. Smith, I. Foster and V. Taylor, “Scheduling with Advance Reservations”, *in proceedings of the IPDPS Conference*, May 2000.

[24] SURFNET: http://www.surfnet.nl/info/en/home.jsp.

[25] Teragrid: http://www.teragrid.org/.