

Revised Slide Window First Algorithm for Advanced Reservation in Optical Grid

¹SantoshM:Sabale; ²GirishV:Chowdhary

³MaheshA:Khandke

Department of Computer Engineering
Dr. Babasaheb Ambedkar Tech. University
Lonere, Raigad, M.S., India

Department of Information Technology
Gharda Institute of Technology
Lavel, Ratnagiri, M.S., India

¹sabale_sm@yahoo.com;
³maheshkhandke@gmail.com
²girish.chowdhary@gmail.com

Abstract

For utilizing distributed resources in optical grid environment advanced reservations play a very crucial role. For applications like the co-allocation of distributed resources and deadline driven applications advance reservations are essential. Also for enhancing capabilities of resource brokers advance reservations play a major role. The objective is to determine a scheduling algorithm that minimizes the blocking probability, i.e. the probability of not scheduling a request within its window, minimizes reservation delay and maximizes network utilization. In this paper, we proposed revision of Slide Window First (RSWF) algorithm. Also the comparison has been done depending on parameters such as reservation delay, blocking probability, link utilization and so on with Slide Window First (SWF) algorithm. Here, RSWF algorithm checks one path at a time for all of the scheduling window slots. If a path cannot be reserved during the scheduling window, the next shortest path is checked. In this, Parallel Dijkstra's algorithm is used for finding the shortest path.

I. INTRODUCTION

Over the past few years it has become evident that local computational resources cannot keep up with the ever increasing demand for processing power. The solution to this problem came in the form of distributed computing, aggregating the power of a multitude of computational resources in one big Grid. This Grid is named after the analogy with the electricity grid, and provides users with on-demand resource usage. Advanced reservation will play a major role in Optical Grid environment. This mechanism guarantees the availability of resources to users at some specified future time. It can ensure the future availability of the Grids heterogeneous resources and help a scheduler to produce better schedules [4].

In this paper, we proposed RSWF algorithm, which attempts to minimize reservation delay and maximize link utilization of optical grid in an advanced reservation scenario. Rest of the paper is organized as follows. section 2 explains the problem description. RSWF algorithm is explained in

section 3. Comparison and simulation results are explained in section 4. Finally, conclusion and future work is described in section 5.

II. PROBLEM DESCRIPTION

Consider a Network Topology Graph $G = (N, L, W)$ where N is the set of nodes, L is the set of links and W is the set of wavelengths supported by each link. A user submits an advance reservation request for a light path between any two nodes on G to the Domain Network Resource Managers (DNRM). Each request R is defined by the following parameters:

$$R = [\text{source node, destination node, } s, e, d, \text{ bandwidth}]$$

Where, d is the reservation duration, and s and e are the starting and ending time of the scheduling window respectively as shown in Figure 1. The time is slotted with a slot size equal to t_0 . The scheduling window defines the time period within which the requestor would like to make a resource reservation. The scheduling window must be bigger than the reservation duration d . Thus the scheduler must check if a path is available during interval $[s + t, s + t + d]$ where $t = 0, 1, 2, \dots, e - s - d$ [7].

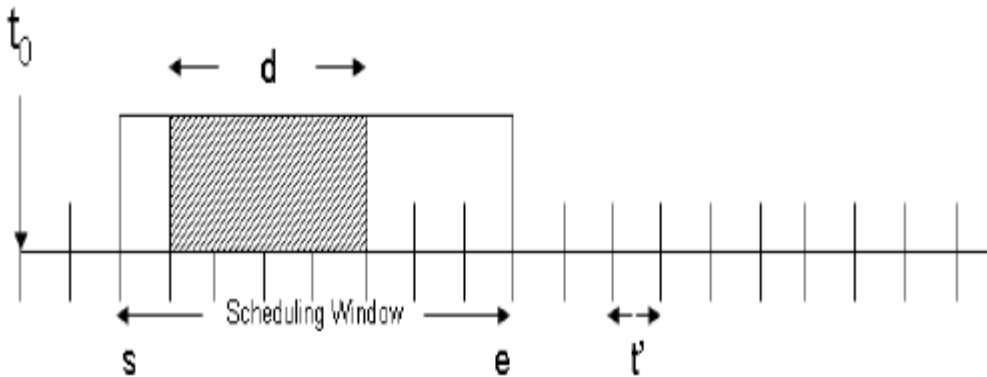


Fig. 1. Scheduling Window.

This is an online scheduling problem because the requests arrive dynamically and for each request R , the DNRM must compute a path and then check if a wavelength on each link of this path can be reserved for duration d within the scheduling window $[s, e]$. The DNRM allocates a wavelength on each link along a path from the source to the destination nodes. If a wavelength along the path for the specified period of time is not available, another path has to be determined. In order to do this, the DNRM maintains a schedule of the reservations called the Reservation Table. It contains all current and future reservations and it is used to search for available resources for new advance reservations [7].

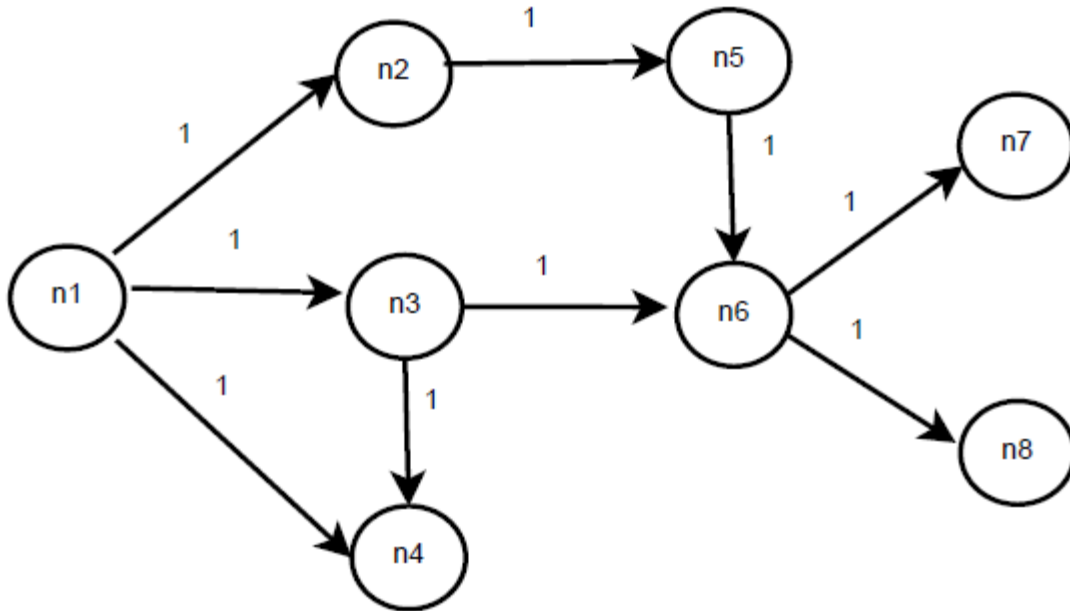


Fig. 2. Example Topology.

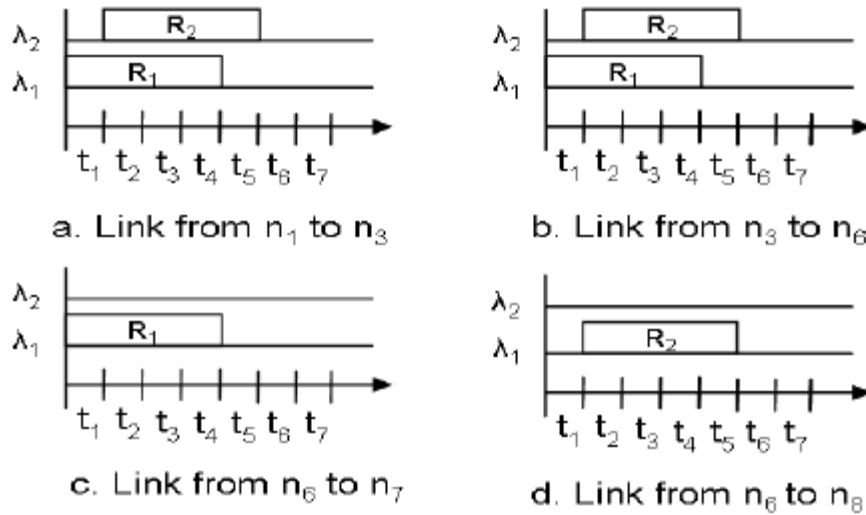


Fig. 3. Reservation Table at t_0 .

Table 1 shows an example of the reservation table for two light path requests for the network shown in figure 2. This is an optical network with 2 wavelengths per fiber link and each link has a cost of 1. Let us assume that at time t_0 two light path requests arrive, $R_1 = [n_1, n_7, t_1, t_8, 4, 1]$ and $R_2 = [n_1, n_8, t_2, t_9, 4, 1]$. We assume that each request requires a bandwidth equal to a wavelength. As there are no other reservations at this time, the links are reserved starting at the beginning of

the scheduling window of each request. A pictorial representation of the reservation table at t_0 is shown in figure 3[7].

Let us assume that a third request arrives at time t_1 for a path between n_1 and n_8 with $R_3 = [n_1, n_8, t_3, t_8, 3, 1]$. Since all the wavelengths along links $n_1 - n_3$ and $n_3 - n_6$ are busy till time t_4 , the shortest path $n_1 - n_3 - n_6 - n_8$ is not available for slots t_3 and t_4 . But due to the large scheduling window, the request can be still accepted for slots t_5, t_6 and t_7 for the same path [7].

At time t_2 a fourth request arrives with $R_4 = [n_1, n_7, t_3, t_5, 1, 1]$. In this case, all the wavelengths along links $n_1 - n_3$ and $n_3 - n_6$ are busy till t_5 and the shortest path $n_1 - n_3 - n_6 - n_7$ is not available for all slots in the scheduling window. So in this case, another path has to be determined. This new path can be a 4-link path i.e. $n_1 - n_2 - n_5 - n_6 - n_7$ and the wavelengths that are

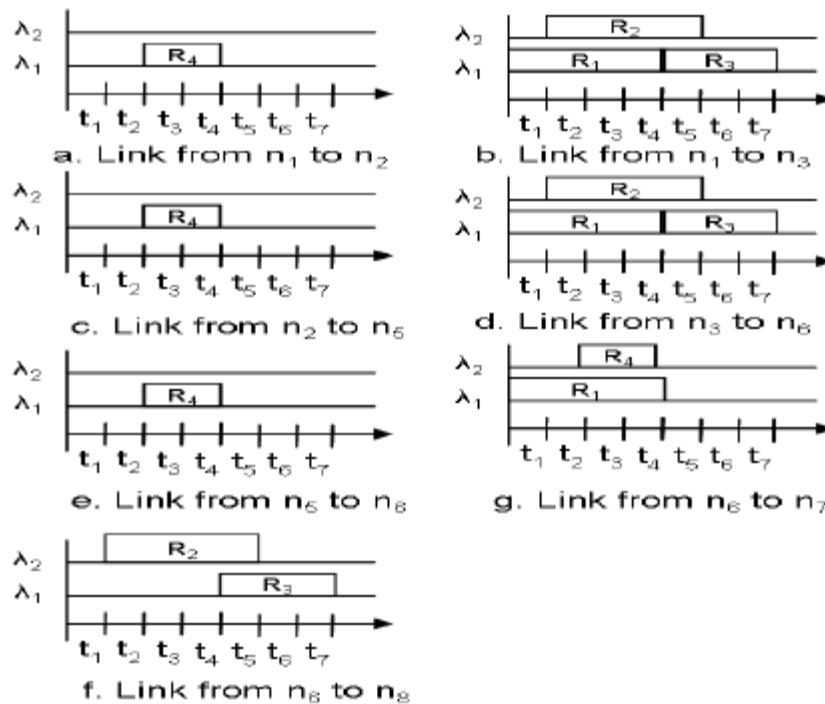


Fig. 4. Reservation Table at t_2 .

TABLE I
RESERVATION TABLE AT t0

Node1	Node2	Wavelength	Start Slot	End Slot	Req ID
n1	n3	λ_1	t1	t4	R1
n1	n3	λ_2	t2	t5	R2
n3	n6	λ_1	t1	t4	R1
n3	n6	λ_2	t2	t5	R2
n6	n7	λ_1	t1	t4	R1
n6	n8	λ_1	t2	t5	R2

reserved are 1; 1; 1 and 2.

The state of the reservation table at time t2 is given in figure 4 [7].

Authors of [7] have discussed Slide Window First (SWF) and Switch Path First (SPF) algorithms. The results of SWF algorithm as compared to the SPF algorithm are good [7]. But the reservation delay needs to be minimized for the SWF algorithm. Also, the blocking probability needs to be minimized.

To overcome this problem, we have proposed revision of SWF algorithm. Here, our focus is to evaluate and compare RSWF algorithm for advance light path scheduling that can be implemented in a DNRM. The main aim of this paper is to find the best scheduling policy for a Grid network resource manager that improves network utilization and minimizes blocking probability by introducing parallel programming in SWF algorithm.

III. REVISED SWITCH PATH FIRST (RSWF) ALGORITHM

In this algorithm, we try to find a free period d starting at s + t, where t = 0, 1, 2, ...e - s - d. If the first shortest path is not free for the required duration during the window, the busiest link defined as the one that uses the maximum number of slots during the scheduling window, is removed from the network topology and the procedure is repeated until either an available path is found or a maximum of k paths is considered. The only difference between SWF algorithm and RSWF algorithm is that Dijkstra's algorithm is implemented parallely [12] in RSWF algorithm.

In parallel Dijkstra algorithm [12], author has maintained two sets: D for all discovered nodes, U for undiscovered nodes At the start, D contains the starting node s, the current estimated best distance from s to v can be denoted with the help of d(v). Initially value of d(v) for all v except s is infinity i.e. d(v)=infinity. The set D grows by adding the node with minimal estimated distance from U into D, and update the current distance of the neighbor nodes of the newly added node, and repeat this procedure until all nodes are in D. Let w(u,v) denote the edge length between adjacent nodes u and v. The pseudo code for Parallel Dijkstra Algorithm is as follows:

- 1) Algorithm 1: Parallel Dijkstra Algorithm
- 2) for all v in G
- 3) d(v) = infinity;

- 4) $D = s$;
- 5) $U = G-s$;
- 6) for all u adjacent to s ;
- 7) $d(u) = w(u, s)$;
- 8) while U is not empty
- 9) Let v be the node from U with minimal $d(v)$;
- 10) $U = U/v$;
- 11) $D = D \cup v$;
- 12) for all u adjacent to v
- 13) if $d(u) > w(u, v) + d(v)$ then
- 14) $d(u) = w(u, v) + d(v)$;

The pseudo code for Revised Slide Window First Algorithm is as follows:

- 1) Algorithm 2: The RSWF algorithm
- 2) function FindPath(Request r , topology t)
- 3) $i = 1$
- 4) while ($i \leq k$) do
- 5) start time = s
- 6) end time = $s + d$
- 7) find shortest path with Parallel Dijkstra's algorithm with propagation delay link cost
- 8) if A path is found then
- 9) while ($endtime \leq e$) do
- 10) if wavelengths are available on all links during start time and end time then
- 11) assign wavelengths, update all tables
- 12) return
- 13) else
- 14) start time = start time + t_0
- 15) end time = start time + d
- 16) end if
- 17) end while
- 18) else
- 19) remove the busiest link during the window from topology
- 20) end if
- 21) $i++$
- 22) end while
- 23) end function

IV. EXPERIMENT RESULTS

GridSim Toolkit [10] is used for conducting simulation run, since it supports advanced reservation. Here, we have taken 44 nodes and 94 unidirectional links. We assume 10 wavelengths on each link and full wavelength conversion at each node.

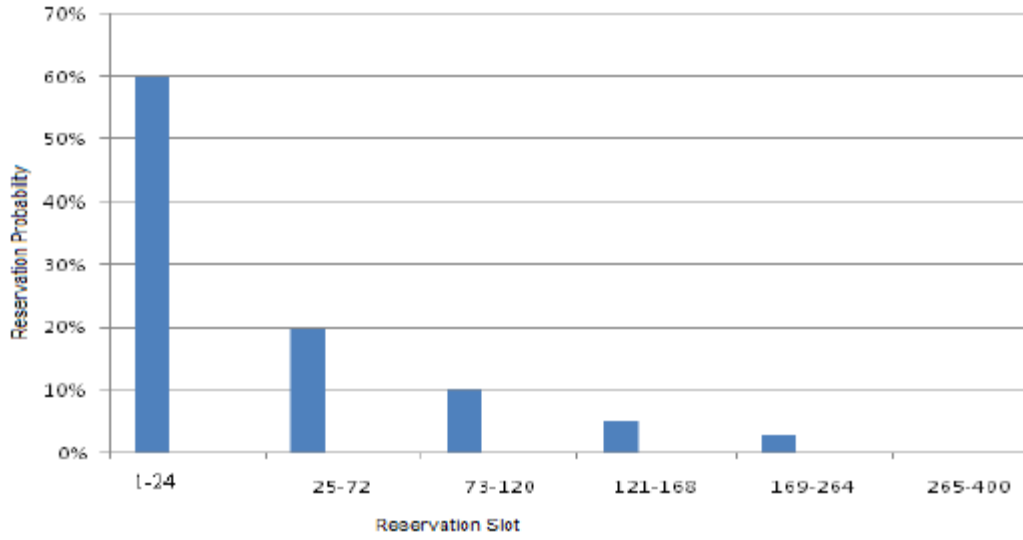


Fig. 5. Discrete Probability Distribution for the intermediate time between the Request Arrival Time and Reservation Time.

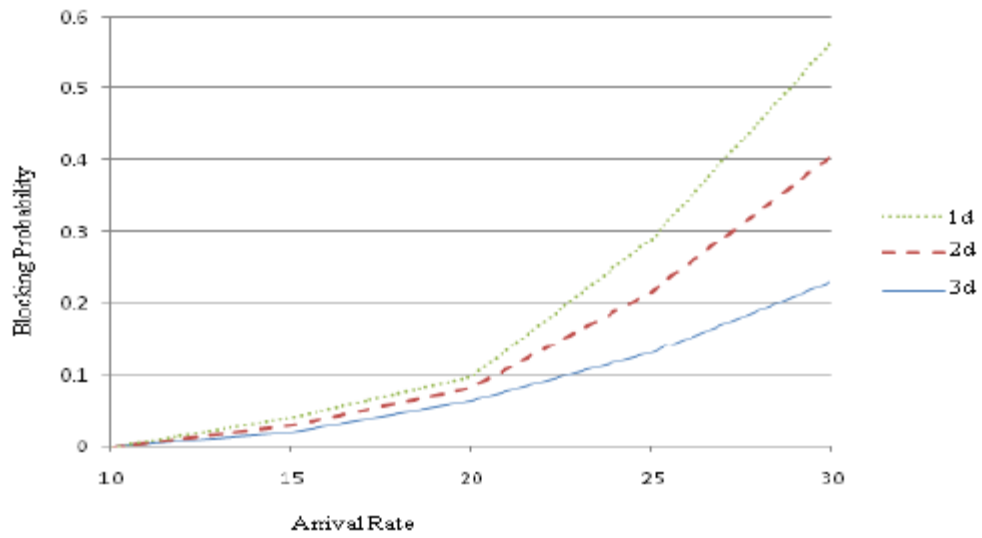


Fig. 6. Flexibility in Scheduling Window.

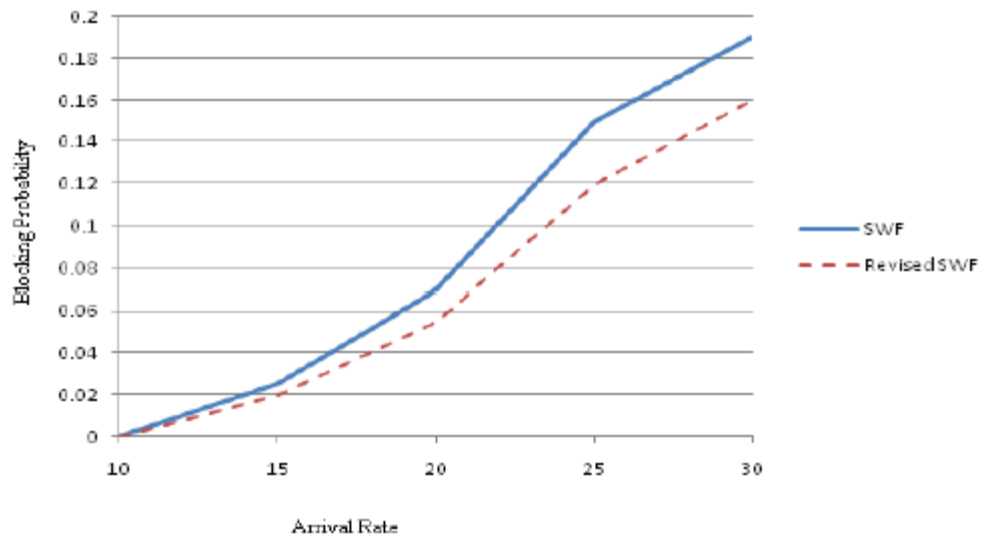


Fig. 7. Arrival Rate vs Blocking probability.

We assume that requests arrive in a Poisson fashion and all requests need to reserve a lightpath with bandwidth equal to

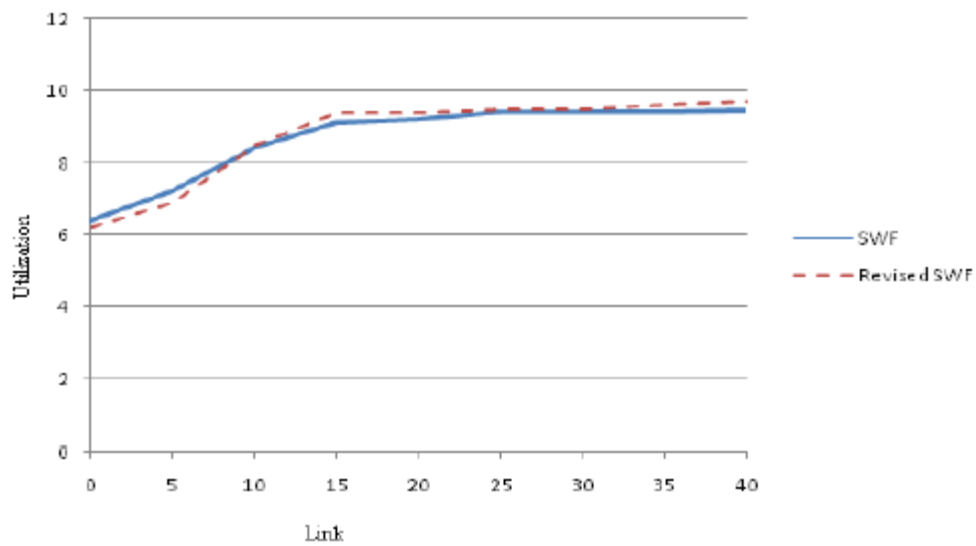


Fig. 8. Link Utilization.

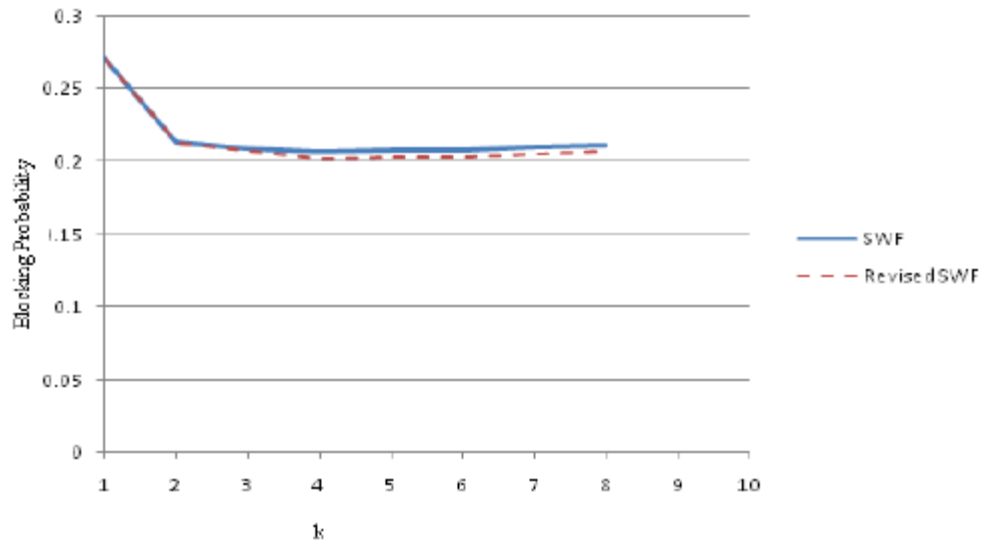


Fig. 9. k vs Blocking Probability.

one wavelength. The duration of a reservation is uniformly distributed. The start times of the request are generated within a window of 400 slots. To simulate a more realistic environment, we have generated the intermediate period between the arrival of the request and the start of the reservation using the discrete probability distribution shown in figure 5. We assume that more requests will be for the reservation slots in the near future i.e next 24 slots and very few requests will be for reservation slots far into the future e.g. after two or three hours. The source and destination nodes for the requested connection are selected randomly using a uniform distribution.

We assume that the scheduling window is twice the reservation duration, i.e., $(e - s) = 2d$, because the results of RSWF shows that using just 1 or 2 units of flexibility improves the performance significantly. Figure 6 gives the blocking probability for different window sizes. The solid curve marked as 1d corresponds to the case $e - s = d$. The dotted line curves correspond to the cases when $e - s = 2d$ and $e - s = 3d$. It can be interpreted that the scheduling window with a width of twice the duration has much lower blocking probability than without any flexibility. Also increasing it further does not improve the performance to a large extent but the delay between the start time of the actual reservation and the start of the window increases. Simulations runs were carried under different network loads, where network load is determined

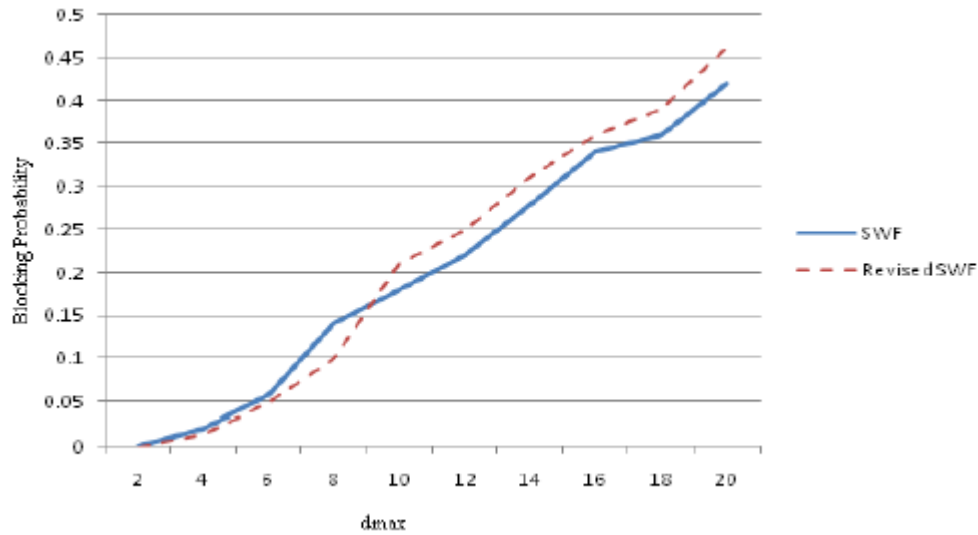


Fig. 10. dmax vs Blocking probability.

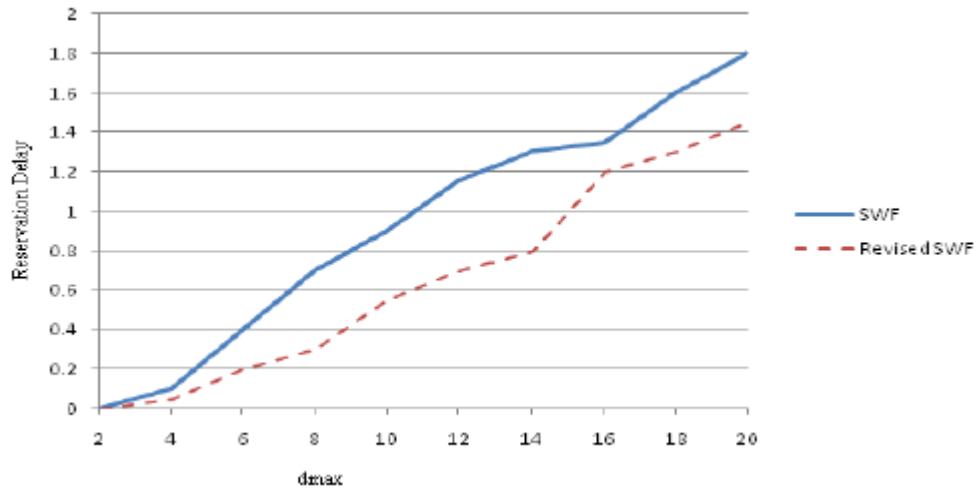


Fig. 11. Reservation Delay vs dmax.

by the request arrival rate and the reservation durations. The parameter of interest is the blocking probability B_p .

Figures 7 show the effect of the arrival rate on the blocking probability. The arrival rate is expressed in terms of number of requests/slot. From the graph, it can be interpreted that RSWF performs slightly better than SWF because it tends to schedule the connections on shorter paths. The link utilization comparison between SWF and RSWF is shown in figure 8. These results are for an arrival rate of 60 requests/slot. The slopes of the curves show how the load is balanced among the links with and without load balancing. The links are sorted in the order of utilization. It is also noted that for most of the links, revised SWF achieves higher utilization than SWF. This

figure shows that for less than 10 links, the utilization of RSWF is low as compared to the SWF. After links exceeds 10, the utilization of RSWF is slightly high as compared to the SWF algorithm. In short approximately 5 percent link utilization of RSWF is increases as compared to the SWF algorithm.

RSWF algorithm tries k alternate paths before blocking a request. To find the optimum value of k , we ran the simulation with different values of k . The results are shown in figures 9. It was observed that the blocking probability decreases initially but then it increases slightly as k increases. As we increase the values of k , longer paths are reserved which indirectly affects

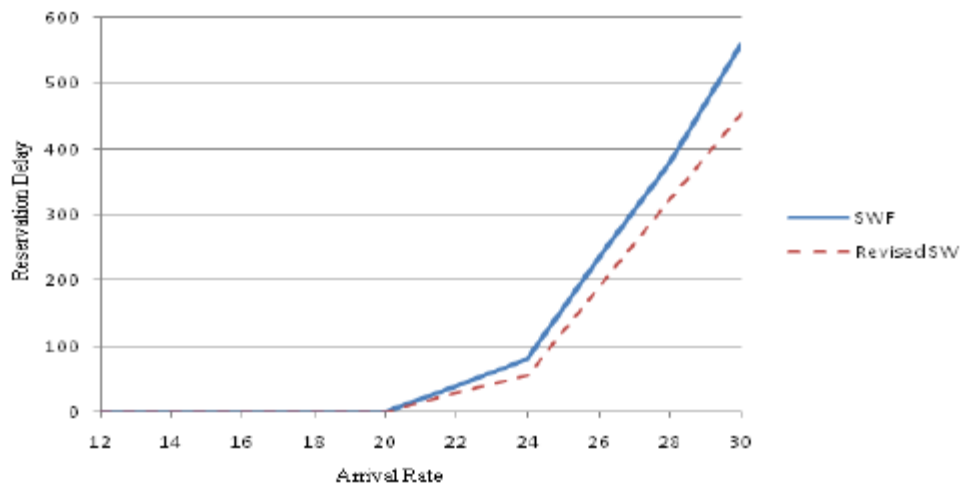


Fig. 12. Average Reservation delay.

the overall blocking probability. When $k=1$, RSWF has same blocking probability as SWF because there are no alternative paths. The value of k with the least blocking probability varies with the network topology. The value of k depends on the number of links in the network but it is independent of the link costs. The optimum value of k was used for each scheme for the rest of the experiments e.g. it is 4 for both SWF and RSWF.

The graphs in figures 10 show the effect of the connection duration d on the blocking probability. The duration is uniformly distributed with a minimum of one slot and a maximum d_{max} . It can be interpreted that the blocking probability increases as d_{max} increases. Here, initially RSWF performs best when d_{max} is below 9. After d_{max} exceeds 9, the blocking probability of SWF is lower as compared to the RSWF.

Figures 11 shows the reservation delay, i.e., the time elapsed from the requested start time s to the time $s + t$ where the reservation was actually made, as a function of d_{max} for both SWF and RSWF. It can be interpreted that RSWF always tries to schedule as close to the start time s of the scheduling window as possible. Revised SWF algorithm has lower reservation delay as compared to the SWF algorithm. Results show that approximately 20 percent delay is minimized with the help of RSWF.

Non-blocking version of the scheduler is also implemented here. In this case, the requests are never blocked and scheduled at the first available time which can be outside the scheduling window. In this case, both the revised SWF and SWF algorithms keep on sliding the window until they find an available path. The results are given in figure 12. In this figure, the average reservation delay is plotted versus the arrival rate for both scheduling schemes. Also the delay increases tremendously to an average of 500 slots which is not desirable. But at lower rates, there is a low percentage of connections scheduled outside the window and also with a low reservation delay. It can be interpreted that the RSWF algorithm gives better results because they tend to reserve the request at a time close to the start of the scheduling window. Obviously in case of a non-blocking scheduler, a stability condition needs to be developed in order to protect the network from becoming unstable, i.e., the reservation delay becomes very large. Such a condition is not necessary for a blocking scheduler, where a request will get blocked if it cannot be scheduled within its requested window [s, e].

Table 2 compares SWF and RSWF algorithm of Optical Grid.

TABLE II
COMPARISON OF SWF AND REVISED SWF ALGORITHM OF OPTICAL GRID

	Slide Window First Algorithm	Revised Slide Window First Algorithm
Reservation Delay	High as compared to RSWF	Low as compared to SWF
Blocking Probability	High as compared to RSWF	Low as compared to SWF
Link Utilization	Less as compared to RSWF	More as compared to SWF
Complexity	High as compared to RSWF	Low as compared to SWF

V. CONCLUSION

Advanced reservation represents an important mechanism in Optical Grid which allows applications to request resources for use at a specific time in the future. For advanced reservation in Optical Grid many algorithms are present.

The RSWF algorithm obtains better results as compared to the SWF. Reservation delay of RSWF is decreases about 20 percent as compared to the SWF. Also, the blocking probability of RSWF is decreases about 10 percent as compared to the SWF. Link utilization of RSWF is increases about 5 percent as compared to the SWF.

VI. FUTURE WORKS

The network utilization can be improved further by offline optimization of the reserved connections that are not in service yet. Also, several failure recovery schemes are present in advanced reservation. Detailed work needs to be done in failure recovery schemes for further improvement of performance.

REFERENCES

- [1] Ahmar Abbas, Grid Computing: A Practical Guide To Technology and Applications, Firewall Media, 2008.
- [2] Ian Foster, Carl Kesselman, Steven Tuecke , “The Anatomy of the Grid”, 2001.
- [3] Ian Foster, Carl Kesselman, Steven Tuecke , Jeffrey M. Nick, “The Physiology of the Grid”, 2002.
- [4] Mark Bartlett, Alan M. Frisch, Youssef Hamadi, Ian Miguel, Chris Unsworth, “Efficient Algorithms for Selecting Advanced Reservations”, IEEE,2006.
- [5] Jawad Ashraf and Thomas Erlebach, “A New Resource Mapping Technique for Grid Workflows in Advance Reservation Environments”, IEEE, 2010.
- [6] Hidemoto Nakada, Atsuko Takefusa, Katsuhiko Ookubo, Makoto Kishimoto Tomohiro Kudoh, Yoshio Tanaka, Satoshi Sekiguchi, “Design and Implementation of a Local Scheduling System with Advance Reservation for Co-allocation on the Grid”, IEEE, 2006.
- [7] Savera Tanwir, Lina Battestilli, Harry Perros and Gigi Karmous- Edwards, ”Dynamic Scheduling of Network Resources with Advance Reservations in Optical Grids ”, 2007.
- [8] Min Zhu, Wei Guo, Shilin Xiao, Anne Wei, Yaohui Jin, Weisheng Hu, and Benoit Geller, ”Availability-Driven Scheduling for Real-Time Directed Acyclic Graph Applications in Optical Grids”, Opt. Comm. Network, 2010.
- [9] Wei Guo, Weiqiang Sun, Weisheng Hu, Yaohui Jin, ”Resource Allocation Strategies for Data-Intensive Workflow-based applications in Optical Grids”, IEEE, 2006.
- [10] GridSim Website. <http://www.gridbus.org/gridsim>, 2008.
- [11] Anthony Sulistio, Rajkumar Buyya, ”A Simulation Infrastructure for Resource Allocation with Advanced Reservation in Global Grids”, 2004.
- [12] Fang Zhou Lin, Nan Zhao, ”Parallel Implementation of Dijkstra’s Algorithm”, 2004.