In-Advance First-Slot Scheduling with Wavelength Conversion for e-Science Applications

Yan Li, Sanjay Ranka and Sartaj Sahni
Department of Computer and Information Science and Engineering
University of Florida, Gainesville, Florida 32611
Email: {yanli, ranka, sahni}@cise.ufl.edu

***Abstract*—For on-demand scheduling of optical networks, it is known that networks equipping a relatively small amount of wavelength converters can provide a comparable blocking probability to that of networks in which each nodes have a wavelength converter. However, the impact of wavelength converters on in-advanced scheduling, especially First-Slot scheduling is not known. In this paper, we conduct a thorough study on the relationship between wavelength converters and First-Slot RWA algorithms using various metrics. We introduce a model for networks with sparse wavelength conversion and employ *EBF* and *KDP* as our RWA algorithms. We evaluate the performances of these algorithms under various topologies and workloads by varying the fraction of routers that have wavelength conversion. Our results show that increasing the fraction of nodes with wavelength converters reduces the blocking probability, but have only marginal effects on requests' average start time. However, both metrics are heavily affected by network topology and traffic load. We also observe that accepting requests greedily, like *EBF* does, yields better performance when the workload is low, but the more conservative approach, *KDP*, is superior when the workload is high. Other contributions of this paper are a carefully selected tie-breaking scheme and a hybrid scheduling algorithm that selects the scheduling strategy based on current workload.**

I. Introduction

Dedicated optical networks have enabled a variety of large-scale scientific applications, called *e-Science*, that are both data intensive and geographically distributed. These applications, generate huge amounts of data (order of terabytes to petabytes) and require these data to be transferred across the network. Examples of *e-Science* applications includes high energy physics [1], astronomy [2] and climate study [3]. The data transfer requests from e-Science applications consist of a lightpath that provides guaranteed bandwidth for certain duration. Depending on how the request's start time is specified, the requests can be classified into two categories: *Fixed-Slot* requests and *First-Slot* Requests. *Fixed-Slot* requests specify a exact start time at which the path should be available, while *First-Slot* Requests describe a time window in the future and expect to be fulfilled as early as possible within this reservation window. Scheduling algorithms for both request types are presented in [4]–[6]. Several reservation systems already have the ability to schedule such requests by reserving a lightpath in an optical network [7]–[10].

In optical network scheduling, the primary concern is routing and wavelength assignment (RWA). Wavelength division multiplexing (WDM) allows multiple lightpaths from different users to share one optical fiber simultaneously. Normally, a feasible lighpath in the optical network has to fulfill the *wavelength continuity constraint*, which forces a single lightpath to occupy the same wavelength throughout all the links that it spans. However, this constraint is relaxed in an optical network that is equipped with wavelength converters. The signals received by a wavelength converter may be transmitted on a different wavelength in the next hop. When every node in the network is equipped with a wavelength converter, the network supports *full wavelength conversion*. When only some of the network nodes have a wavelength converter, the network supports *sparse wavelength conversion*.

Although the design and implementation of a full-conversion scheduler is relatively straightforward, the high cost and the added latency introduced by wavelength converters make sparse-conversion more attractive in practice. Existing research on *On-Demand* scheduling has shown that wavelength converters have the potential to improve blocking performance significantly and that it is necessary only for a relatively small fraction of the nodes to have a wavelength converter to achieve blocking performance comparable to that of full wavelength conversion [11]–[13]. Since on-demand scheduling reserves a path for a fixed time slot, this result applies directly to the Fixed-Slot scenario. The impact of wavelength conversion on First-Slot scheduling has been evaluated in the full-conversion scenario [6]. The focus of the current paper is to understand the benefits sparse wavelength conversion on First-Slot scheduling.

We present a new network model that that can emulate the existing full-conversion algorithms when only a subset of nodes have a wavelength converter. We demonstrate the utility of this approach using the *Extended Bellman-Ford(EBF)* [14] and *k-Alternative Path(k-Path)* algorithms. We choose the *EBF* and *k-Path* algorithms for our study because these two algorithms, respectively, represent the two major classes of First-Slot algorithms that have been proposed for first-slot scheduling. *EBF* guarantees to schedule a request whenever there is a feasible path [6]. The *k-Path* algorithm schedules a request only when this can be done using a path from a set of k precomputed paths. The *k-Path* algorithm is easy to implement and is very fast. As a result, this algorithm is commonly used for both Fixed-Slot and First-Slot scheduling [4], [5], [12]. We evaluate the algorithms on 3 performances metrics: blocking probability, average start time and scheduling overhead. Blocking probability, which measures the ratio of blocked requests to the number of scheduled requests, is the primary metric used to evaluate a scheduling algorithm. Average start time, which presents how early the requested lightpath is available, is of special importance in the First-Slot scenario. Scheduling overhead, which compares algorithms according to their computation costs, is an important metric for the algorithms practicality.

Our primary goal is to study the impact of wavelength converters on First-Slot scheduling and to analyze the temporal behavior of typical RWA algorithms in the context of sparse wavelength conversion. Extensive experimental results are presented in this paper. According to our test results, increasing the fraction of nodes with wavelength converters is of greater value for blocking performance in relatively low traffic cases than in the high workload cases. However, the average start times are almost unaffected by wavelength converters ratio except for the marginally improvement in small topologies.

Another key consideration in this paper is to explore the advantages and disadvantages of the 2 scheduling strategies represented by *EBF* and *k-Path* respectively. Intuitively, always accepting a request whenever there is a feasible lightpath in the network, which is done by *EBF*, should provide better performance than limiting the search for a feasible path to a small set of candidate paths as is done in *k-Path*. However, our results show that this statement is only true when the overall workload is small compared to network capacity. *EBF* often schedules a request on a longer path than used by the *k-Path*. So, when the workload is high, the additional resources utilized by *EBF* to accept a request negatively impacts the acceptance of future requests. Our results clearly demonstrate these tradeoffs: *EBF* performs better when network capacity is ample for the requested workload, but *k-Path* outperforms when the traffics congest the network. This observation leads us to propose a hybrid approach that automatically switches between the two algorithms based on current network traffic.

We also study different tie-breaking approaches when multiple paths are feasible and the impact of different tie-breaking schemes on overall performance in the presence of sparse wavelength conversion. A *slack* tie-breaking scheme is proposed and its performance relative to other widely used strategies is analyzed.

The rest of the paper is organized as follows. Our network model and related data structures are given in Section II. The RWA algorithms used in this paper are detailed in Section III. The experimental environment and results analysis are presented in Section IV. In Section V, we describe related work. Finally, we conclude in Section VI.

II. Network Model and Data Structures

A. Network Model

The topology of a optical network with sparse conversion is represented as a graph $G = (V, E, W)$ where V is the set of optical switches or routers, E is the set of optical links and W is the number of wavelengths supported by each link. Each node n in V is associated with a boolean function $F(n)$, which is true if and only if the node is equipped with wavelength converter. To emulate the full-conversion algorithms, we first convert the above graph into a new graph $G' = (V', E')$. To map node set G to G', for a node $n \in V$, if n equips a wavelength converter (i.e. F(n) is true), a corresponding node n' will be inserted into V'. If F(n) is false, W pseudo-nodes $(n'_1, n'_2, ..., n'_w)$ will be inserted. W is the number of wavelengths defined in G. For a link $l \in E$, if e connects 2 nodes with converters, it will be mapped to a link l' in E'; else, l will be mapped into W pseudo-links $(l'_1, l'_2, ..., l'_w)$ and each pseudo-link stands for a specific wavelength that carried in l. In the extended model, l'_i is incident to n'_i iff l is incident to n in the original graph and l'_i and n'_i is their i_{th} pseudo copy respectively.

Figure 1 shows a example of a 4-node ring topology. The network contains 2 wavelength converters, nodes A and B. Each link carries 2 wavelengths. Its extended presentation, shown on the right side, contains 6 nodes and 7 links. Each non-converter nodes are split into 2 pseudo-nodes(nodes {C1, C2} for node C and {D1, D2} for node D). Each optical link that is incident to a non-converter node is split into 2 pseudo-links and each pseudo-link stands for one individual wavelength(link(B, C1) and (B, C2)for link (B, C) for example). Two pseudo-nodes that are adjacent in the extended model must fulfill 2 conditions: 1)their corresponding nodes are adjacent in the original graph, and 2) they are either converter nodes or their wavelength index matches.

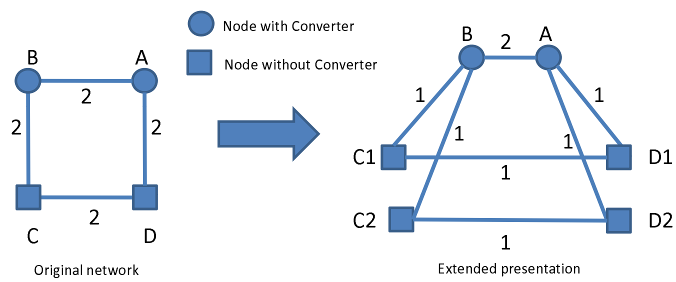


Fig. 1. Extended Network Model

The extended graph G' is equivalent to the original graph G.Every feasible lightpath in G has a corresponding path in G' and vice versa. The wavelength continuity constraint is also preserved in the extended model. If node n has no wavelength converter in G, every corresponding pseudo-node n'_i in G' is incident by and only by the pseudo-links of wavelength index i. Hence, we can directly apply the RWA algorithms that originally designed for full-conversion/no-conversion networks to the sparse conversion scenario with little adaptation.

When scheduling a request, if the source or destination nodes are extended to multiple pseudo-nodes, the RWA algorithm needs to check every corresponding pseudo-source-destination pairs. So at most W^2 rounds of the original algorithm are needed in the extended model, where W is the maximum number of wavelength carried by a link in the network. Also, during the model extension, the graph size can increase by at most W times for its link and node number. Therefore, an algorithm's computation time on the extended model G' is bounded by a constant ratio(a polynomial of W) of the computation time for that algorithm in full-conversion scenarios, which runs on the original network G. So, the adapted algorithm has the same asymptotic complexity as its original version.

In this paper, we employ *Extended Bellman-Ford* algorithm and *k-Alternative path* algorithm as our RWA algorithms. The detail of the algorithms will be presented in Section III.

B. Data Structures

When developing a bandwidth reservation system, one must decide on the representation of time. The options are to either consider time as divided into equal size slots as is done in [4], [15], [16] or to consider time as being continuous as in [14], [17], [18]. For the reasons of space efficiency and results accuracy as stated in [5], we use the continuous time model in this paper. In this model, the status of each link l is maintained using a Time-Bandwidth list (TB list), $TB[l]$, that is comprised of tuples of the form (t_i, b_i), where t_i is a time and b_i is a bandwidth. The tuples in a TB list are in increasing order of t_i. If (t_i, b_i) is a tuple of $TB[l]$ (other than the last one), then the bandwidth available on link l from t_i to t_{i+1} is b_i. When (t_i, b_i) is the last tuple, a bandwidth of b_i is available from t_i to ∞. Consider the link shown in Figure 2. The graph is a pictorial representation of the bandwidth available on this link as a function of time. So, for example, a bandwidth of 5 is available from time 0 to time 1 and the available bandwidth from 2 to 3 is 4. The corresponding TB list is $[(0, 5), (1, 2), (2, 4), (3, 5), (4, 1), (5, 5)]$.

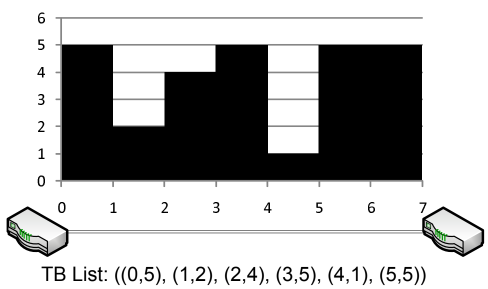


Fig. 2. TB List

All requests are submitted to an centralized scheduler for scheduling. Each request wants to establish a path in the network with some capacity and for some duration. The request also specifies a reservation window $[ST, ET]$, where the path should be available and only available within this window for the requested duration. File transfer requests are characterized by a 6-tuple (s, d, BW, Dur, ST, ET) where s and d is the source and destination location of the path to be established; BW is the number of request wavelengths and Dur is the expected duration of the path. ST and ET specify the start time and end time of the reservation window.

With the definition of both TB List and user request, we define Start-Time List(ST List) on each link in the network to incorporate the TB List and request info together for the start time computation. A ST list is comprised of pairs of the form (a, b) with the property that for any $t \in [a, b]$, the wavelength of this link has bandwidth BW available from t to $t + Dur$. The ST List for each wavelength can be easily computed by scanning the corresponding TB list in linear time. [14] has shown that if we collect all the a values inside $[ST, ET]$ in every link, the earliest start time t for current request is one of the a_is.

III. Routing and Wavelength Assignment Algorithms

The RWA algorithm for First-Slot problem usually contains 3 steps:

1) Identify the earliest start time.
2) Find the shortest path that provides such start time.
3) Assign the wavelength.

As the wavelength assignment are independent from the first 2 steps, most RWA algorithms handle this step separately. In our paper, *EBF* and *k-path* algorithm proceed the first 2 steps. A wavelength assignment strategy called *Least Conversion Assignment* is explained in Section III-D.

A. Extended Bellman-Ford Algorithm

Extended Bellman-Ford algorithm [14] applies the Bellman-Ford shortest path algorithm [19] to the ST lists on links that may connect source and destination. The key steps of the algorithm are as follows:

1) Let st(k, u) represent the union of the ST lists for all lightpaths from vertex s to vertex u that have at most k edges. Clearly, $st(0, u) = \emptyset$ for $u \neq s$ and $st(0, s) = [0, \infty]$. Also, $st(1, u) = ST(s, u)$ for $u \neq s$ and $st(1, s) = st(0, s)$. For $k >= 1$, the following recurrence can be derived:

$$\begin{aligned} st(k, u) &= st(k-1, u) \cup \\ &\quad \{\cup_{v,(v,u) \in E} \{st(k-1, v) \cap ST(v, u)\}\} \end{aligned}$$

2) Construct the list $st(n-1, d)$, which gives the start times of all paths from s to d that have bandwidth BW available for a duration Dur. If $st(n-1, d)$ is not empty, a in its first (a, b) pair is the earliest start time for this current source/destination pair, denoted as est_i.
3) For all possible source/destination pairs, find the minimum est_i as the earliest start time for the request and the corresponding source/destination pair (s_i, d_i) are recorded.
4) Remove the links that can not provide the requested bandwidth at the earliest finish time.
5) Run Breath-First Search [19] on the extended model and find a shortest route from the s_i to d_i, map the route back to the original graph.

The complexity of intersection and union operation is linear to the length of the current ST List. For each iteration of constructing $st(k, d)$ for $st(k-1, d)$, we needs to compute the ST List for each link and each computation takes $O(L)$ time, where L is the length of the longest st list. Since the construction iterates at most $N - 1$ times, the complexity of the extended Bellman-Ford algorithm is $O(N * E * L)$, where N and E is the number of nodes and links in the graph.

B. k-Alternative Path Algorithm

The *k-Alternative path* algorithm is extended from the shortest path algorithms routing used by $OSPF$ in Internet routing. Recognizing that an OSPF-like algorithm may fail to find a feasible path in a network that has a feasible path, k-Path algorithm generates additional disjoint paths with the hope that one of the additional paths will be feasible. By constructing

ST List on each path, it is easy to decide which path provides earliest start time. If the all generated path is infeasible, the request is reject.

The k paths can be either fixed or dynamically generated. k-Fixed path are computed for each node pair before the first scheduling begins according to some given link costs. The k-dynamic path are computed for every request according to current network status. Usually, the most congest link will have large cost so as to be avoid in the path computation. [20] have shown the k-Dynamic paths can provide a much larger network throughput than k-Fixed paths, especially when the traffic load is relatively large for the network's capacity. So, in this paper, we adopt the this k-Dynamic paths algorithm and call it *KDP*.

C. Breaking the Ties in Path Selection

When multiple paths can satisfied a user's request, a tie-breaking scheme is needed to select one of them as the actual scheduled path. The most straightforward scheme breaks ties based on *first-fit(FF)* strategy or *Shortest Path(SP)* strategy. *First-fit* strategy terminates the path selection once a succeed path is found. *Shortest-path* strategy chose the path with minimum hop number. Both strategies are widely used in non-conversion and full-conversion scenarios, but using them directly in Sparse wavelength conversion needs careful consideration as these paths are computed on extended models.

In the extended model, the shortest path does not necessarily corresponds to the shortest path in the original graph for same source/destination pairs. Figure 3 gives an example of a 4-node ring with only 2 converters, nodes A and C. Each link consists 2 wavelengths. At some conjuncture, the available wavelengths are shown in extended graph on the right side. For a request of 1 wavelength of capacity from node B to A. Two candidate paths are available: $B_1 \rightarrow C \rightarrow D_1 \rightarrow A$ and $B_2 \rightarrow A$. Assume that both path provides the same start time, The *First-Fit* tie-breaking scheme will choose the pair (B_1, A) as source and destination when the node-pairs are checked in lexical order and the *Shortest Path* tie breaking scheme will choose the 3-hop path as it is the one available B_1 to A.

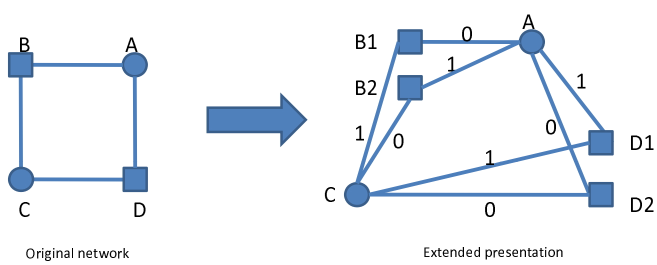


Fig. 3. Extended Network Model

A tie-breaking scheme that choose the shortest path by examining all source-destination pairs, rather than the first succeed pair, would solve the problem. However, considering only the path length would not be enough in the context of sparse conversion. In networks without wavelength conversion, shorter path is more likely have common wavelengths, therefore less likely to block the requests. However, the present of wavelength converters reduces the correlation between path length and path capacities due to the elimination of the wavelength continuity constraint. Hence, in terms of load balancing, choosing longer path with higher capacity may also reduce the potential congestion by alleviate the traffic in the short paths. In this paper, instead of using *First-Fit* strategy in original *EBF* and *KDP* algorithm, we employ a *slack* tie-breaking scheme that selects the path that at most *h* hops longer than the shortest path but have the most free wavelengths among all the paths that have fewer or equal hop counts. We call these variant *EBF-S*and *KDP-S*.

Intuitively, *h* should not be too large or the benefit of load balancing would be canceled by the waste of link capacities on longer paths. In Section IV, we will comparing the *First-Fit* scheme with *Slack* scheme and perform a numeric analysis on the choice of *h* in different scenarios.

D. Wavelength Assignment

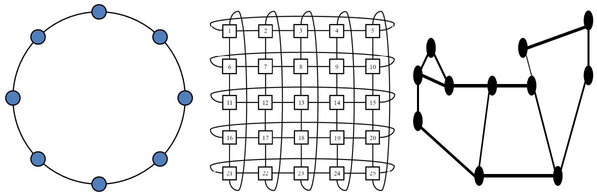
With the presence of wavelength converters, wavelength assignment becomes less important in optical routing. However, as wavelength conversion contributes a considerable delay in the optical transmission [21], a proper wavelength assignment would potentially reduce such overheads.

For those links that connect to a non-converter nodes, the wavelength assignment is quite straightforward, as the specific wavelength has been already identified with the lightpath during the path selection process. However, for those links connect two wavelength converters, we apply the *least conversion assignment* rather than the *first-fit* algorithm used in most RWA algorithms. In *least conversion assignment*, the link that connects 2 converters will use the same wavelength either as its previous hop or as it next hop unless neither wavelengths is not available. For instance, let $l \in E$ connects two wavelength converters, w_i be the wavelength assigned for the previous link on the path and $\{w_n\}$ be the set of common wavelength of both l and l's next link. If w_i is available on l, we assign this w_i to the lightpath, else we assign any wavelength in $\{w_n\}$ if $\{w_n\} \neq \emptyset$. If neither is available, a random available wavelength in assigned. This strategy can always guarantee a feasible solution while avoiding the unnecessary conversions.

IV. Experimental Evaluation

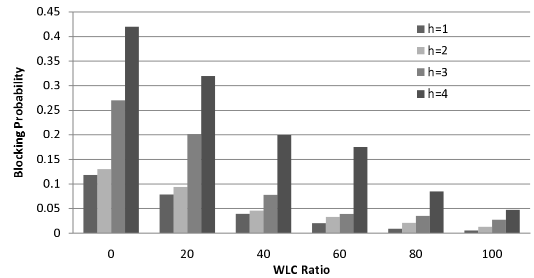
A. Experimental Framework

In this section, we measure the performance of the scheduling algorithms described in Section III and analyze how wavelength converters affects the algorithms' performance in various scenarios. We compare the tie-breaking schemes in Section III-C to show the effectiveness of our *slack* strategy; We analyze the performance of RWA algorithms for 3 metrics: blocking probability, average start time and execution time. In our experiments, blocking probability is measured by ratio of rejected requests comparing to the total submitted request. Average start time is measured by the average delay between the reservation window's start time for a job and its actual start time. Execution time measures the convergence speed of each algorithm and how it varies with network size and workloads. We also proposed a self-adaptive algorithm switching strategy that dynamically choose the suitable algorithm according to the

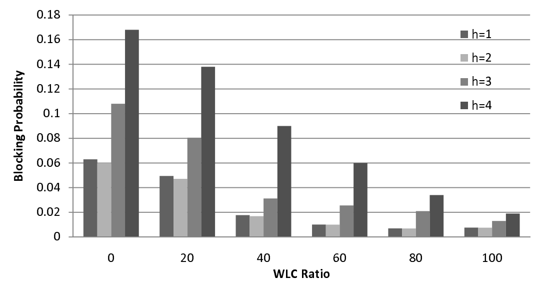


(a) Ring (b) Mesh-Torus (c) Abilene

Fig. 4. Network Topologies



(a) Ring



(b) Random

Fig. 5. Different h values for different topologies. Network Traffic Load: $\alpha = 0.05$

current workload. The performance of this switch strategy is tested in various scenarios.

To simulate a e-Science backbone, we use a 10-node ring topology, a 25-node mesh-torus topology, a real work 11-node Abilene network (Figure 4c) and several randomly generated topologies. Each link is assumed to carry 10 wavelengths. For randomly generated topologies, we set the out-degree of each node to be random integers between 3 and 7. To ensure network connectivity, the random network has bidirectional links between nodes i and $i+1$ for every $1 \leq i < n$, where n is the number of nodes. During the results analysis, we found that the test data from Ring and Abilene topology are very similar to each other, while Mesh-Torus and random topology also results in almost same observations. Thus, in this paper, we only present the results from Ring and Random topology to avoid redundant diagrams. The full analysis on data from all four topologies can be found in our technical report [22].

Besides ratio of nodes that have converters, their placement in the network is also an important consideration and extensive studies have been performed [23]–[26]. In this paper, we do not employ any specific wavelength converter placement policies for following reasons: 1) most converter placement strategies are closely related to traffic distribution in the network, but the traffics pattern are not precisely known at the network design time. 2) we assume that topology and converter placement are predefined. The scheduler cannot make changes. Instead, a simple placement strategy is used in our evaluation. A node is capable of wavelength conversion with probability of q independent of the other nodes. The number of wavelength converters in a network of N nodes is binomially distributed with an expectation of Nq. In our simulation, we use *WLC Ratio* to indicate the percentage of nodes that are equipped with wavelength converters. We will compare the algorithms performance under different *WLC Ratio* to plot the impact of wavelength converters. Although a simple strategy is applied here, our expectation is that a performance comparison between different approaches should be applicable even when a more sophisticated placement is used.

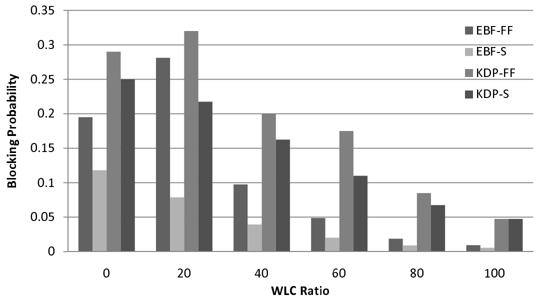
File transfer requests are synthetically generated. Each request is described by the 6-tuple $(s_i, d_i, BW_i, Dur_i, ST_i, ET_i)$. The source and destination nodes for each request were selected using a uniform random number generator so that the workload is distributed uniformly among different node pairs. Without loss of generality, we assume each request asks for a capacity of only 1 wavelength. The duration is uniformly distributed within a range of 100 to 500 time units. The arrival of the request follows a Poisson distribution with rate α. The reservation window starts at some time after the requests' arrival. As the results are relatively insensitive for this lag, we arbitrarily chose it as 100 time units. The length of the window is randomly select from 2 to 4 times of the request duration.

We assume that the requests arrive in a Poisson process for each source/destination pair with an arrival rate α. Following the experimental setting in [11], α is picked in the range from 0.01 requests/time unit to 0.1 requests/time unit for each node pair. So, for example, with a arrival rate of 0.05 requests/time unit, one run of our experiment on a 100 node random topology would process approximately $5 * 10^5$ requests during the tests which lasts 1000 time units. All our experiment assume that we start with no load i.e., no existing scheduled transfers.

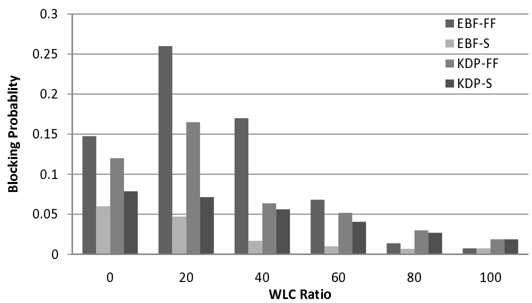
B. Slack *Tie-Breaking Scheme*

Recall that in Section III-C, to select the best candidate path, we consider all the paths that are at most h hops longer than the shortest path. Hence, to evaluate the performance of these heuristics, we must first decide the value of the h. Figure 5 explore how h's value influence the blocking performance of $EBF-S$ in various topologies. In the small topology like 8-node ring, $h = 1$ provides the best performance, while in the 100-node random network, the $h = 2$ case is marginally better than $h = 1$. Also, large h values like 3 or 4 actually deteriorate the performance as the link capacity is wasted by choosing long paths. In our test, $h = 1$ is also the best choice for Mesh-Torus and Abilene. Similar results (not presented here) were also observed for $KDP-S$. In the following tests, we chose $h = 1$ in for Ring, Abilene and Mesh-Torus topologies and $h = 2$ for random network.

Figure 6 depicts the wavelength converters' impact on blocking performance by varying the wavelength converter ratio for various topologies. *EBF-S* and *KDP-S* use the simple *First-Fit* tie-breaking scheme but *EBF-S* and *KDP-S* employ *slack*

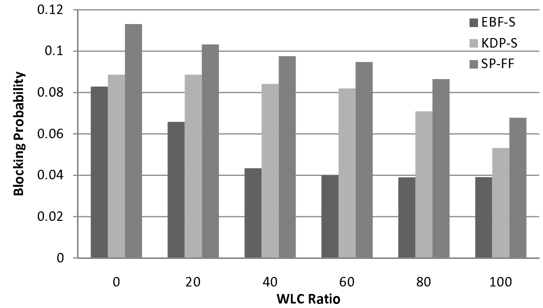


(a) Ring

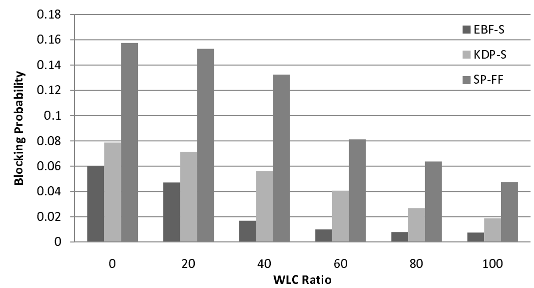


(b) 100-node Random

Fig. 6. Benefit of *slack* tie-breaking scheme in various topologies. Network Traffic Load: $\alpha = 0.05$.



(a) Ring



(b) 100-node Random

Fig. 7. Blocking Probability vs. Wavelength Converter Ratio in various topology with low traffic load.

scheme. We observe that the algorithms with *slack* scheme work much better than the algorithms using *FF* scheme in all cases. Thus, choosing longer path in presence of excess capacity benefits the blocking performance for sparse wavelength conversion. Another observation is that when *WLC Ratio* equals 20%, the blocking probability of *First-Fit* algorithms is worse than not having any converters. This phenomenon can be explained as follows: when no wavelength converter is available, many requests are rejected due to lack of continuous wavelength, but the accepted request are more evenly distributed among all wavelengths and long lightpaths are less likely to be established. When a small number of nodes are equipping with converters, the traffic scheduled by *First-Fit* strategy are more likely to use those long paths between with wavelengths of lower index, as shown in Section III-C. The additional benefit from having 20% converters is not enough to cover the degradation due to the wasted capacity of long paths. Although this degradation can be compensated by either inserting more converters, as plotted in Figure 6, using the *slack* scheme is obviously more effective and economical. We also notice that the improvement brought by the *slack* scheme in Ring and Abilene topology is not as much as in mesh-torus and random topology. This is consistent with the conclusion in [11], which states that wavelength conversion can help more in the topologies with more divergence and connectivity, as more variants in the paths are available.

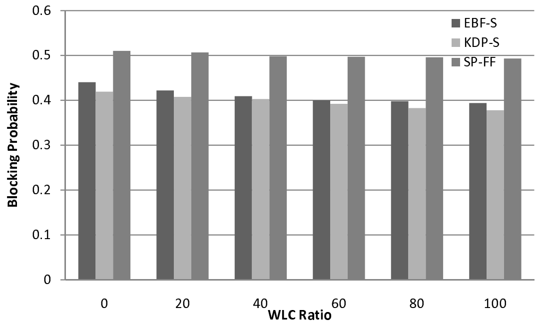
In the test for both Figure 12 and Figure 6, the network traffic load is set to a moderate degree: $\alpha = 0.05$. However, similar results are also observed under different workloads.

C. Blocking Probability

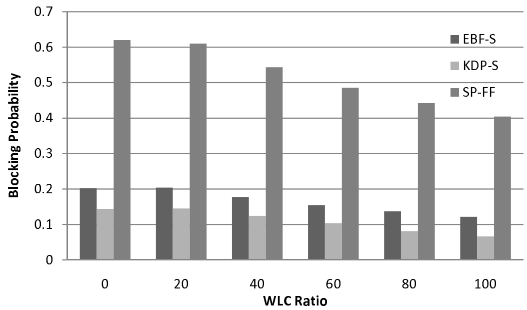
In this section, the blocking performance of *EBF-S* and *KDP-S* are evaluated and compared the *Fixed-Shortest Path* routing algorithm. For *Fixed-Shortest Path* routing, no tie-break is specified as only one path is available. However, as *First-Fit* wavelength assignment is applied, *Fixed-Shortest Path* routing is denoted as*SP-FF* in our diagrams. Figure 7 and Figure 8 depict the how the blocking probabilities changed with wavelength converter ratio in Ring, Mesh-Torus, Abilene and Random-100 topology. Figure 7 present the results obtained when the network's traffic load is relatively low: $\alpha \in [0.01, 0.05]$, while Figure 8 present the result when workload is relatively high: $\alpha \in (0.05, 0.1]$.

From Figure 7 and Figure 8, we note that Increasing the wavelength converter ratio can decrease the blocking probabilities for all algorithms. However, the improvement is also dependent on the network's traffic load. When network traffic load is relatively low, *EBF* only needs about 40% of wavelength converters to provide a satisfactory blocking performance, but the blocking probabilities of *KDP* and *SP* decreases more gradually as the increase of WLC Ratio. When traffic load is high, increasing the wavelength converters has only marginal improvement on all algorithms. This can be explained as follows. *EBF* explores the network more thoroughly for an available path than *KDP* and *SP*. With a small amount of converters, *EBF* is able to satisfy the traffic demands but *KDP* and *SP* can not. However, when traffic load is high, majority of blocking occurs due to lack of link capacities but not the availability of continuous wavelengths. Hence, increasing wavelength converters has little impact in heavy loads.

We also note that in all topologies. *EBF-S* and *KDP-S* algorithms can achieve a much smaller blocking probability comparing to *SP-FF* algorithm. In simple topologies like Ring (Figure 7a and 8a), *EBF-S* and *KDP-S* leads *SP-FF* for about 2-5% on blocking probability, while in those more complex

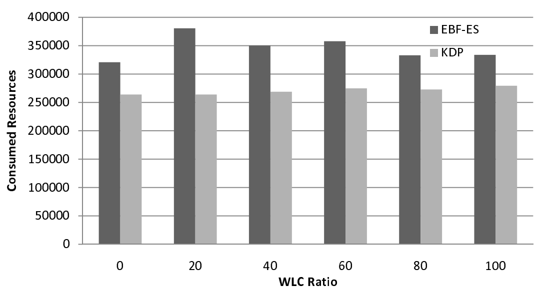


(a) Ring

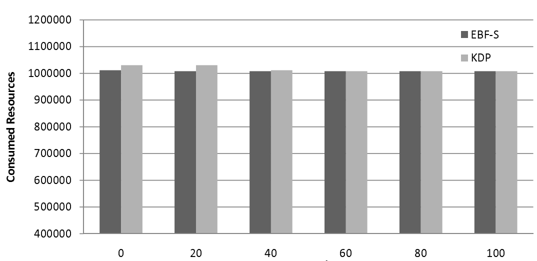


(b) 100-node Random

Fig. 8. Blocking Probability vs. Wavelength Converter Ratio in various topology with high traffic load.



(a) Low Workload

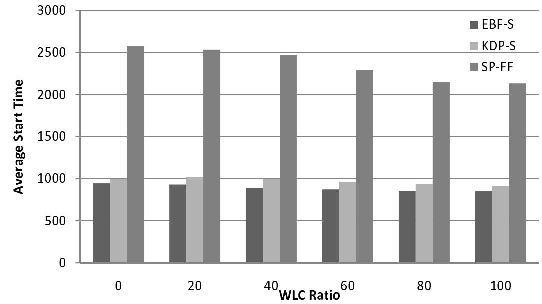


(b) High Workload

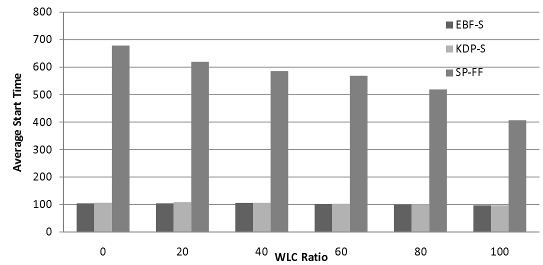
Fig. 9. Total resource consumption in a 100-node random network under different workload.

topologies like random network (Figure 7b and 8b), the advantages are doubled. This is consistent with the results in [12] that shortest-path routing is less likely to be improved due to the small number of alternative lightpaths.

Another important observation is that *EBF* outperforms *KDP* in low traffic workloads(Figure 7), but *KDP* leads *EBF* in high traffic workloads(Figure 8). This can be explained as follows: *EBF* tries to find any possible path for current request if it



(a) Ring



(b) 100-node Random

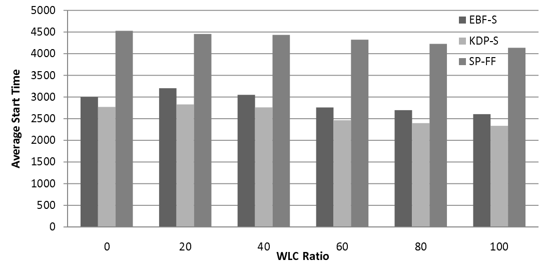
Fig. 10. Average Request Start Time vs. Wavelength Converter Ratio in various topology with low traffic load.

exists, but *KDP* only tests no more than k paths. When the long-term traffic load is low, the network's capacity is ample to accommodate most requests, the *EBF* that acts greedily would accept more requests than *KDP*. However, when traffic load is high, limiting the routes to only those short paths and reject some long paths would definitely benefit the schedule of future requests. In that case, the more conservative *KDP* would provide a better performance in the long run.

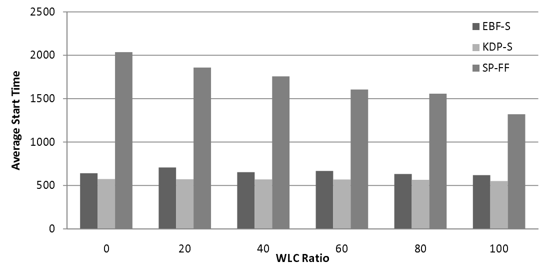
This observation can also be supported by the fact that *EBF* actually consumes network bandwidth faster than *KDP* does. Figure 9 depicts the total amounts of link capacities that *EBF* and *KDP* consume under different workloads. The total resource amounts are defined as $\sum_{p \in RLP}(Dur(p) \cdot length(p))$, where RLP is the set of all established lightpaths, $Dur(p)$ is lightpath p's duration and $length(p)$ is p's hop count. We see that, for the same request set, *EBF* consumes more link capacities than *KDP* in low workload cases, as *KDP* rejects those long-path requests while *EBF* accepts them. When the workload is high, *KDP*, which rejects some early long-path requests, substantially accepts more requests in the long run. Therefore, the total amounts of link capacities both algorithms consumes are almost equal in high workload.

D. Requests' Average Start time

In the all-optical routing area, blocking probability is always the primary concern, but for this special case of First-Slot scheduling, the availability of earlier start time may also be an important metric to evaluate a scheduler's performance. Figure 10 and Figure 11 present the influence of wavelength converter ratio on the requests' average start time. Similar with the previous section, Figure 10 shows the start time performance in the low traffic case and Figure 11 shows the test result in high workload cases.

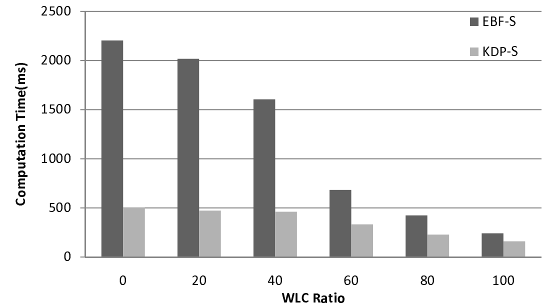


(a) Ring

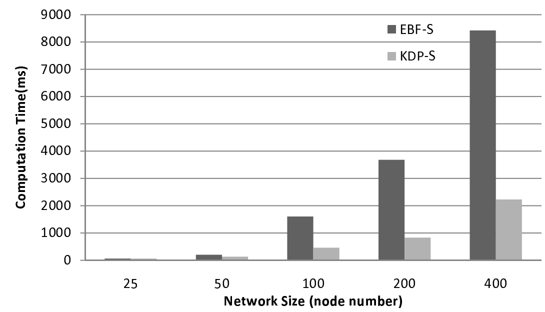


(b) 100-node Random

Fig. 11. Average Request Start Time vs. Wavelength Converter Ratio in various topology with high traffic load.



(a) CT vs. WLC Ratio



(b) CT vs. Network Size

Fig. 12. Average computation time of *EBF-S* and *KDP-S*.

To exclude the impact of blocked requests on the average request start time, we set the reservation window for each request to a large enough time interval such that every request will be accepted at some time within the window. From the above figures, we can observe that increasing wavelength converters has positive impacts on the requests' start time, but the improvements are not as obvious as the impact on blocking performance, especially for *EBF-S* and *KDP-S*. In Ring topology, the improvement from 0% WLC ratio to 100% WLC ratio is only about 5% in low traffic load case, and 15% in high traffic load cases. In random topology, the improvements are almost negligible.

We also note that *EBF-S* and *KDP-S* lead the average start time over *SP-FF* in all case. The advantages are larger in random topology than in Ring topology and they increase as the workload increase. This shows that *EBF-S* and *KDP-S* have the ability to schedule the requests in a more parallel way than *SP-FF*. When the network capacities are ample for the requests, *EBF-S* and *KDP-S* provide much faster schedules. When the traffic congests the network, those forthcoming requests have to wait for previous requests to finish due to lack of network capacity, which reduce *EBF-S* and *KDP-S*'s advantages.

Similar to the results in previous section, *EBF-S* again outperformed *KDP-S* in low workload cases on average start time, and *KDP-S* gains better performance for conservative reservation strategy in the heavy workload. When the networks' capacity is relative ample comparing to the requested workload, *EBF-S* has the ability to start the job earlier than *KDP-S*, but their performances are very close in large topologies. When the work network is under congestion by high request rates, *KDP-S* leads the average start time in all topologies and its advantage are more observable in large networks.

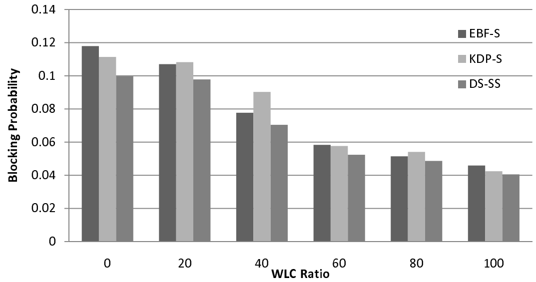
E. Scheduling Overhead

In this section, we discuss about the scheduling overhead of our RWA algorithms. Figure 12 presents comparisons on the scheduling overhead of *EBF-S* and *KDP-S* with various WLC ratio and network size. These results show that *EBF-S* is about 2-5 times slower than *KDP-S* and its computation time grows faster than *KDP-S* with the increase of network size. The execution time decreases with the increase in WLC ratio, as the size of extended network is smaller. However, even for a 400-node topology, *EBF-S* can schedule a request averagely within several seconds. This should be acceptable in most scenarios.

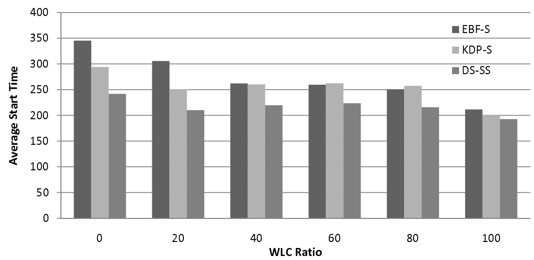
In summary, equipping the network with some wavelength converters do improve the blocking performance, but adding more converter after certain threshold does not bring more benefits. Meanwhile, the traffic load and network topology also cast great influence on the blocking probability. For the average start time, we found that the effect of wavelength converter ratio is minor for both *EBF-S* and *KDP-S*, but the traffic load and network topology have more evident influences on this metric. Comparing two algorithms, *EBF-S* generally performs better in the low traffic cases, but *KDP-S* is a better choice when the traffic load is heavy. *KDP-S* is faster than *EBF-S* in all case, but in even in the worst case, *EBF-S* can still schedule the requests in an acceptable speed.

F. Algorithm Switching Strategy

We find out in Section IV-C and Section IV-E that the greedy approach *EBF-S* has better performance when the traffic load is comparably light, while the conservative approach *KDP-S* works more effectively in the high workload case. In this section, we propose a self-adaptive algorithm switching strategy that automatically chooses *EBF-S* or *KDP-S* according to current traffic load in the network.

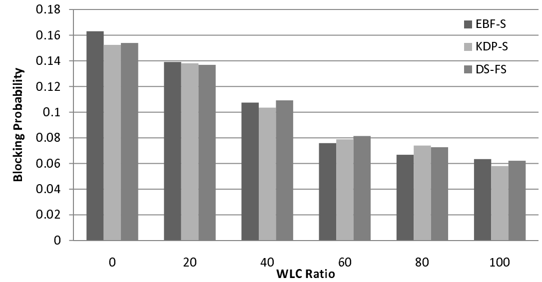


(a) Blocking Performance

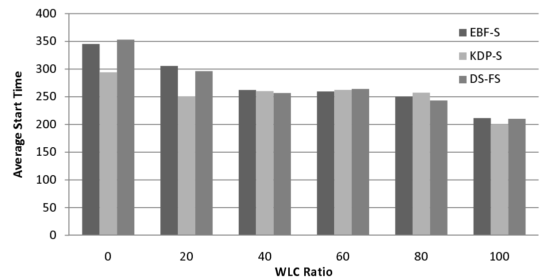


(b) Start Time Performance

Fig. 13. The performance of algorithm switching strategy in Slow Traffic Pattern Switching.



(a) Blocking Performance



(b) Start Time Performance

Fig. 14. The performance algorithm switching strategy in Fast Traffic Pattern Switching.

The main idea of the switch strategy is as follow. The time domain is divided into equal-length time slots. The scheduler runs both algorithm simultaneously on each request, the choice of whose result to apply in current time slot is made at the beginning of this slot, assuming current traffic has the similar pattern as in last slot. The performances of the candidate algorithms can be evaluated by either their blocking performance, or average start time of the requests scheduled in the last slot, or a combination of the two. Comparing the statistical performance of both algorithms in the last slot, the algorithm that performed better in the last slot takes effect in the current slot. If it does not perform as good as the other one in current slot, the schedule will switch to its alternative at the beginning of the next slot.

To evaluate the performance of our algorithm switch strategy, we designed 2 scenarios: slow traffic pattern switching(*STPS*) and fast traffic pattern switching(*FTPS*). In our test, we assume the request arrival rate α is randomly selected from the range $[0.01, 0.1]$ and the length of each time slot is 10 time units. In *STPS* scenario, the arrival rate α changes its value with 50% probability every 100 time unites. In *FTPS* scenario, the arrival rate changes its range with 50% probability every 10 time units.

Figure 13 and Figure 14 depicts the blocking and start time performance of our switch strategy in both scenarios in a 100-node random topology. *DS-SS* algorithm stands for the switch strategy worked in the *STPS* scenario, while *DS-FS* algorithm stands for the switch strategy worked in the *FTPS* scenario. The results show that the hybrid algorithm worked in *STPS* scenario, *DS-SS*, have the best performance on both metrics, as the history information from last slot predict the current traffic pattern quite accurately. On the other hand, the performance of *DS-FS* algorithm which worked in *FTPS* scenario degraded dramatically due to the increasing probability of inaccurate predictions in *FTPS* scenario. In some cases, *DS-FS* even provides the worst performance. So, when the traffic pattern remain static or changed infrequently, our algorithm switch strategy can provide a pretty good performance by synthetic the merits of both algorithms. However, when the traffic pattern is changing frequently, the switch strategy does not guarantee any improvement comparing to either *EBF-S* or *KDP-S*.

V. Related Work

Generally, bandwidth reservation systems operate in one of two modes [5]:

(a) In **on-demand** scheduling, bandwidth is reserved for a time period that begins at the current time.
(b) In **in-advance** scheduling, bandwidth is reserved for a time period that begins at some future time.

On-demand scheduling is a special case of in-advance scheduling: the time interval between the request's arrival and its actual starting time is zero. On-demand scheduling, which is typically supported by Multiple Protocol Label Switching (MPLS) [27] at layer 3 and by Generalized MPLS (GMPLS) [28] at layers 1 and 2, is supported by CHEETAH, DRAGON, and UCLP. Typical algorithms for on-demand scheduling can be found in [29]–[31]. In-advance scheduling is supported by GeantII, OSCARS, USN and Enlightened. The corresponding algorithms are described in [4], [6], [17].

A summary of various Fixed-Slot and First-Slot RWA algorithms for optical networks with no converters may be found in [5]. The *Extended Bellman-Ford* algorithm was first proposed in [14] to solve the First-Slot problem without wavelength converters whereas the *k-Alternative Path* algorithm was originally proposed to solve the Fixed-Slot problem. [16] adapted the *k-Alternative Path* algorithm to the First-Slot scenario. In [6], both algorithms are adapted for networks with wavelength converters at every node.

The impact of wavelength converters in all-optical routing has be widely studied. [11] showed that for certain topologies and fixed-path routing, sparse wavelength conversion is almost as effective as full wavelength conversion. [12], [21] have investigated the performance of the *k-Alternative Paths* algorithms in the presence of wavelength converters. Their focus is the blocking probability of sparse wavelength conversion for on on-demand scheduling. [6] considers in-advance scheduling using a continuous time model and evaluates the various algorithms' blocking performance for full wavelength conversion and low workload.

VI. Conclusion

In this paper, we examined the impact of sparse wavelength conversion on First-Slot scheduling. We proposed a new network model to emulate the full-conversion algorithms in sparse conversion networks. Using this model, we conducted extensive experiments to assess the impact of wavelength converters on First-Slot RWA algorithms' performance. This assessment used three metrics: blocking probability, average start time and scheduling overhead. Our experiments have indicate that increasing wavelength converters has positive impact on blocking performance, but very little impact on the availability of earlier start times. Meanwhile, as most improvements are achieved by having no more than 60% nodes with converters, deploying too many wavelength converters may not be worth the additional cost. We also proposed a *Slack* tie-breaking scheme when multiple feasible paths are available. This tie-breaking scheme is shown to have much better performance than the traditional *First-Fit* or *Shortest-Path* tie-breaking schemes. The comparisons between *EBF* and *KDP* also lead to the conclusion that accepting requests greedily would provide better performances in low traffic load case, but rejecting some requests with long path can be a superior strategy when workload is high. An algorithm switching strategy that adapts the scheduling algorithm as the current workload changes is proposed. When the network traffic pattern changes slowly, this strategy has considerable advantage over static algorithms. Overall, our results show that adding a small number of wavelength converters may have a limit positive impact on First-Slot scheduling. However, this impact should be carefully weighted against the additional cost.

Acknowledgment

This work was supported, in part, by the National Science Foundation under grant 0312038 and 0622423. Any findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF. The work was also supported in part by a grant from UltraHiNet and Florida High Tech Corridor.

References

[1] "International thermonuclear experimental reactor," http://www.iter.org.

[2] "Joint institute for very long baseline interferometry," http://www.jive.nl/.

[3] "The earth system grid," http://www.earthsystemgrid.org/.

[4] R. Guerin and A. Orda, "Networks with advance reservations: The routing perspective," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, 2000, pp. 118–127.

[5] E. Jung, Y. Li, S. Ranka, and S. Sahni, "An elvaluation of in-advance bandwidth scheduling algorithms for connection-oriented networks," in *Proceedings of International Symposium on Parallel Architectures, Algorithms, and Networks*, 2008.

[6] E. Jung, Y. Li, S. Ranka, and S. Sahni, "Performance evaluation of routing and wavelength assignment algorithms for optical networks," in *13th IEEE Symposium on Computers and Communications*, 2008.

[7] "On-demand secure circuits and advance reservation system," http://www.es.net/oscars.

[8] Geant2, http://www.geant2.net.

[9] Enlightened Computing, http://www.enlightenedcomputing.org/.

[10] DOE UltraScienceNet: Experimental Ultra-Scale Network Testbed for Large-Scale Science, http://www.csm.ornl.gov/ultranet.

[11] S. Subramaniam, M. Azizoglu, and A. K. Somani, "All-optical networks with sparse wavelength conversion," *IEEE/ACM Trans. Netw.*, vol. 4, no. 4, pp. 544–557, 1996.

[12] X. Chu and B. Li, "A dynamic rwa algorithm in a wavelength-routed all-optical network with wavelength converters," in *INFOCOM*, 2003.

[13] M. Kovacevic and A. S. Acampora, "Benefits of wavelength translation in all-optical clear-channel networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 868–880, 1996.

[14] S. Sahni, N. Rao, S. Ranka, Y. Li, E.-S. Jung, and N. Kamath, "Bandwidth scheduling and path computation algorithms for connection-oriented networks," in *Sixth International Conference on Networking (ICN'07)*, 2007, p. 47.

[15] L. Burchard, "On the performance of networks with advance reservations: Applications, architecture, and performance." in *Journal of Network and Systems Management*, 2005.

[16] S. Tanwir, L. Battestilli, H. Perros, and G. Karmous-Edwards, "Dynamic scheduling of network resources with advance reservation in optical grids," in *International Journal of Network Management*, vol. 18, 2008, pp. 79–105.

[17] N. S. Rao, S. M. Carter, Q. Wu, W. R. Wing, M. Zhu, A. Mezzacappa, M. Veeraraghavan, and J. M. Blondin, "Networking for large-scale science: Infrastructure, provisioning, transport and application mapping," in *Proceedings of SciDAC Meeting*, 2005.

[18] N. S. V. Rao, W. R. Wing, S. M. Carter, and Q. Wu, "Ultrascience net: Network testbed for large-scale science applications," *IEEE Communications Magazine*, 2005.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. New York: The MIT Press, 2001.

[20] K. Rajah, S. Ranka, and Y. Xia, "Scheduling bulk file transfers with start and end times," in *6th IEEE International Symposium on Network Computing and Applications*, 2007, pp. 295–298.

[21] X. Chu, J. Liu, and Z. Zhan, "Analysis of sparse-partial wavelength conversion in wavelength-routed wdm networks," in *INFOCOM*, 2004.

[22] Y. Li, S. Ranka, and S. Sahni, "Tech report of cise uf: In-advanced first-slot scheduling with spare wavelength conversion for e-science applications," 2009.

[23] X. Chu, B. Li, and I. Chlamtac, "Wavelength converter placement under different rwa algorithms in wavelength-routed all-optical networks," *IEEE Transaction on Communications*, vol. 51, no. 5, pp. 607–617, 2003.

[24] S. Subramaniam, M. Azizoglu, and A. K. Somani, "On the optimal placement of wavelength converters in wavelength-routed networks," in *INFOCOM*, 1998, pp. 902–909.

[25] J. M. Yates, M. P. Rumsewicz, and J. P. R. Lacey, "Wavelength converters in dynamically-reconfigurable wdm networks," *IEEE Communications Surveys and Tutorials*, vol. 2, no. 2, 1999.

[26] H. Zang, R. Huang, and J. Pan, "Designing a hybrid shared-mesh protected wdm networks with sparse wavelength conversion and regeneration," 2002.

[27] U. Black, *MPLS and Label Switching Networks*. Prentice-Hall Pub., 2002.

[28] N. Yamanaka, K. Shiomoto, and E. Oki, *GMPLS Technologies*. CRC Taylor Francis Pub, 2006.

[29] Q. Ma, P. Steenkiste, and H. Zhang, "Routing high-bandwidth traffic in max-min fair share networks," in *ACM SIGCOMM*, 1996, pp. 115–126.

[30] Q. Ma and P. Steenkiste, "On path selection for traffic with bandwidth guarantees," in *5th Intl. Conf. on Network Protocols (ICNP)*, 1997, pp. 191–204.

[31] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," in *IEEE JSAC*, 1996, pp. 1228–1234.