Advance lightpath reservation for WDM networks with dynamic traffic

T. Daniel Wallace,[1] Abdallah Shami,[1,*] and Chadi Assi[2]

[1]*Department of Electrical and Computer Engineering, The University of Western Ontario, London, Ontario, Canada*
[2]*Faculty of Engineering and Computer Science, Concordia University, Montreal, Quebec, Canada*
[*]*Corresponding author: ashami@eng.uwo.ca*

Received February 1, 2007; revised May 2, 2007; accepted May 9, 2007; published June 26, 2007 (Doc. ID 79573)

Advance reservation is a topic that is rarely discussed within the domain of wavelength division multiplexed (WDM) networks. However, for many emerging applications in the telecommunication and/or grid computing industries, a demand for a high bandwidth communication channel as well as a guarantee on resource availability certainly exists. Such applications include: remote surgery, remote experimentation with teleobservation capabilities, teleconferencing, and bulk transfers. We present what we believe to be a new model for reserving advance lightpath requests in a centralized system. This model attempts to "migrate," i.e., move previously reserved lightpaths to candidate wavelengths in order to lower the system's blocking probability. We have tailored different lightpath migration algorithms to address two specific network objectives: (1) minimize the number of hops a new request traverses after migration, and (2) minimize the number of migrated lightpaths. © 2007 Optical Society of America

OCIS codes: 060.4250, 060.2330.

1. Introduction

A new architectural paradigm referred to as grid computing is currently gaining wide acceptance by engineers and computer scientists in both business and research communities as the future of real-time parallel distributed computing. Similar to the idea of a power grid that supports numerous regions with reserves of electricity, computational grids can distribute a range of services such as data repositories, increased processing power, and additional network bandwidth. Since its inception in the 1990s, most of the advancements made in grid computing have come from the community of e-Science. The definition of e-Science is usually regarded as global distributed collaboration for scientific computing enabled by the Internet [1]. Presently, there are a number of widespread e-Science grid networks that are dependent on long distance backhaul communication links for information and resource sharing. The Biomedical Informatics Research Network (BIRN) is an example of a grid system in action. BIRN supplies a grid infrastructure to three separate U.S. based test-bed projects for national collaborations in biomedical engineering [2]. Another example is the Large Hadron Collider (LHC) Computing Grid Project, which aims to build and maintain a data storage and analysis infrastructure for the world's largest physics laboratory, the European Organization for Nuclear Research (CERN) [3]. Currently, CERN is home to construction of the largest scientific instrument on the planet. When completed, it is estimated that the LHC will annually produce over 15 petabytes of data, which will in turn be accessed and analyzed globally. Another grid community is the George E. Brown Network for Earthquake Engineering and Simulation (NEES) [4] program, which is composed of a diverse group of individuals and organizations whose goal is to study the effects of impact and aftermath of seismic events on common societal structures such as buildings, bridges, roads, etc.

Since many e-Science applications generate massive amounts of data that can range into the region of terabytes or even petabytes, the problem of transportation and network connectivity can become paramount. Currently, bulk transfers such as these necessitate postal couriers for the efficient and reliable delivery of electronic data since the public Internet only offers a best-effort service that can result in weeks of exhaustive uploading. For many e-Science communities, this has motivated the

1536-5379/07/070913-12/$15.00

deployment of privately owned high-speed data networks (e.g., virtual private networks) employing wavelength-division multiplexed (WDM) infrastructures that can provide guarantees in quality of service (QoS). Unfortunately, network infrastructure and bandwidth provisioning (on its own) does not solve the problem of resource reservation and coallocation [5,6], which is a premier requirement for many e-Science applications, where coallocation is the process of reserving multiple resources for a single job.

In this paper, we introduce the advance reservation dynamic lightpath establishment (ARDLE) problem by developing a new model that supports optical networks with dynamic traffic and advance lightpath demands. Our new advance lightpath reservation model uses traditional routing and wavelength assignment (RWA) methods for WDM networks operating under centralized control. However, the performance of this new model can be improved by carrying out lightpath migration on existing reservations and then rerouting connection requests that were initially rejected. Lightpath migration should increase network utilization and ultimately lower blocking probability since resources are managed more efficiently. Furthermore, since reservations are made in advance, rerouting will not affect transmission and thus, there will be no disruption period during migration.

The rest of this paper is organized as follows. Section 2 presents an overview of the literature related to advance lightpath reservation as well as provides background with reference to lightpath migration in WDM networks. Section 3 proposes two lightpath migration algorithms for WDM networks employing advance reservation. Finally, Section 4 evaluates the performance of the proposed algorithms through discrete event simulation, and Section 5 concludes this study and discusses future work.

2. Literature Review and Related Background

2.A. Advance Lightpath Reservation

Although WDM networking is a well-studied technology, only recently has the idea of advance reservation become an area of interest among researchers. In terms of static lightpath demands, advance reservation can be solved as an optimization problem by means of either a mixed integer linear program (MILP) or global approximation schemes such as tabu search (TS) or simulated annealing (SA) [7,8]. Turning towards networks with dynamic traffic, it is typical to approach the problem simply through heuristics rather than optimization due to a continuous change in state information. For instance, Veeraraghavan *et al.* [9] describe a new scheduling algorithm called varying-bandwidth list scheduler (VBLS) that takes advantage of the delay insensitive characteristics of bulk file transfers. VBLS analyzes the available bandwidth on a channel at a given time and determines what portion of a file it can transmit in a single time slot. In this way, if the necessary resources are unavailable at the requested time, they are reserved in advance for the next available time slot. Unfortunately, since this approach assumes the existence of static point-to-point network connections, it does not scale very well. In a more pervasive approach, Zheng and Mouftah introduce the idea of reserving lightpaths (i.e., network connections) using time slots in advance over optical networks in [10,11]. They later proposed a model for reserving lightpaths in advance operating in a distributed network with dynamic traffic in [12]. In this work the authors describe a first fit RWA algorithm for reserving future time slots on the wavelengths of interconnected links along a path.

2.B. Lightpath Migration for Immediate Connections

In WDM networks, lightpaths are bound to the wavelength continuity constraint, which specifies that a lightpath cannot perform wavelength switching at intermediate nodes. Although it is possible to perform wavelength conversion, it is an expensive procedure and the technology is not yet mature. However, due to the wavelength continuity constraint WDM networks can become underutilized. By preventing wavelength conversion, the likelihood of a new connection request finding a wavelength continuous route from its respective source to destination will decrease as the load increases. In attempt to solve this problem, networks can perform lightpath migration by rerouting existing lightpaths to alternate wavelengths albeit maintaining their original path.

The basic operations of lightpath migration are wavelength retuning (WR) and move to vacant (MTV), both of which have been presented in [13]. While WR moves the wavelength of an existing lightpath to a candidate wavelength by maintaining its path, MTV reroutes a lightpath to a vacant path. Although MTV can preserve the transmission of the old route of a lightpath while the new path is being established, i.e., no disruption period since the new route is vacant, it is considered more complex than WR since a new path needs to be computed. On the other hand, WR can be considered advantageous because a new route does not need to be calculated and thus the complexity of the algorithm is reduced, however, when WR is employed, there will be an expensive disruption period if the path along the new wavelength is not vacant. It has therefore been recognized that both methods should be used in tandem with one another by moving a lightpath to a vacant wavelength on the same path. This method of lightpath migration is referred to as move-to-vacant wavelength retuning (MTV-WR).

Since it may be necessary to move more than one lightpath in order to accommodate a new connection request and moving a single lightpath is still considered an expensive procedure, the authors in [13] proposed *parallel* MTV-WR, which is a method for minimizing the weighted number of existing lightpaths needed to be migrated. In [13], it is assumed that lightpaths are setup and maintained for an undetermined amount of time, or in other words all connections are established on demand. When a new connection request arrives, an initial phase attempts to route the request without migrations, however, if no path can be found, a second phase is executed in an attempt to migrate existing lightpaths in order to free up resources for the new request. The work in [13] was later re-examined by authors in [14] by combining both phases of the rerouting algorithm into one so as to reduce the computational complexity.

3. Advance Lightpath Migration

In this section we present a *parallel* MTV-WR algorithm for advance lightpath reservation. To use this algorithm, the model assumes an advance lightpath request has already been attempted and failed. Furthermore, we have developed our model with the intention of addressing two specific network objectives. The first objective is to minimize the total number of hops a new connection request traverses. This function attempts to lower future blocking probability by only utilizing the minimum number of resources needed for a new connection request. In the second objective, we want to minimize the total number of lightpaths that need to be rerouted in order to accept a new connection request. In this way, the time required by a centralized controller to make the appropriate configuration changes will also be minimized. Both objective functions involve a graph transformation phase followed by a cost labeling phase. The route with the least cost is chosen by performing Dijkstra's shortest path algorithm on each of the labeled subgraphs. Those lightpaths that traverse the path with least cost will be migrated to candidate wavelengths.

3.A. Minimize the Number of Hops a Request Traverses After Migration

Our first objective is to find the minimum number of hops a new connection request must traverse in order to make an advance reservation. To achieve this objective we first construct the graph $\cup_{\lambda \in \Lambda} G(N^{\lambda}, L^{\lambda})$ from the set of wavelengths Λ, where N^{λ} and L^{λ} are the set of nodes and links on wavelength λ, respectively. Next, we will define the proper notation required to label the cost of each link in the graph. For this we first develop a method of evaluating whether a new connection request v overlaps any lightpaths u from the set of existing lightpaths U, in both time and physical domains. Given the link $(i^{\lambda}, j^{\lambda}) \in L^{\lambda}$ and $\lambda \in \Lambda$, the function $Q(i^{\lambda}, j^{\lambda}, u, v) = u \in U$ if a new connection request v is overlapping in time with an existing lightpath u on the link $(i^{\lambda}, j^{\lambda})$ and $Q(i^{\lambda}, j^{\lambda}, u, v) = \text{NULL}$ if u is not overlapping in time with v on link $(i^{\lambda}, j^{\lambda})$. Therefore, we define the function that determines if a new connection request v and an existing lightpath $u \in U$ overlaps in time on link $(i^{\lambda}, j^{\lambda})$ by

$$Q(i^{\lambda},j^{\lambda},u,v)=\begin{cases} u & \text{if } u \text{ and } v \text{ overlap in time}, R(i^{\lambda},j^{\lambda},u)=u, u\in U, \quad (i^{\lambda},j^{\lambda})\in L^{\lambda} \\ \text{NULL} & \text{otherwise} \end{cases}. \tag{1}$$

In Eq. (1), the function $R(i^{\lambda},j^{\lambda},u)=u$ if lightpath u uses link $(i^{\lambda},j^{\lambda})$ and $R(i^{\lambda},j^{\lambda},u)=\text{NULL}$ if lightpath u does not use link $(i^{\lambda},j^{\lambda})$.

Since we are also interested in possibly moving existing lightpaths to alternate wavelengths, we must also identify not only those lightpaths that overlap in time but also those which are retunable (i.e., can be moved to an alternate wavelength while retaining the original path), thus we define $M(u)$, $u\in U$ as the migration function: $M(u)=\text{NULL}$ if the set of retunable wavelengths for lightpath u is empty, and $M(u)=\lambda'$ if u can be migrated to an alternate wavelength along its original path where λ' is the smallest index in the set of retunable wavelengths. To determine λ', we must first define Λ'_u, which is the set of retunable wavelengths for lightpath u, i.e.,

$$\Lambda'_u=\{\lambda'\in\Lambda: \forall u' Q(P(u,j)^{\lambda'},P(u,j+1)^{\lambda'},u,u')=\text{NULL}, j=1,\ldots,H(u), u'\in U\}, \quad u\in U. \tag{2}$$

In Eq. (2), we also introduce the functions H and P, where $H(u)$ is equal to the total number of hops lightpath u traverses and $P(u,j)$ is equal to the jth node along the route of lightpath u, respectively. Therefore, $P(u,j)^{\lambda'}$ would then refer to the jth node along the route of lightpath u on wavelength subgraph λ'. The migration function is now formally defined as follows

$$M(u)=\begin{cases} \min\{\lambda'\in\Lambda'_u\} & \text{if } \Lambda'_u\neq\text{NULL}, \quad u\in U \\ \text{NULL} & \text{otherwise} \end{cases}. \tag{3}$$

To correctly implement lightpath migration in advance, it will be necessary to test if an existing lightpath has begun to use its reservation. Testing the status of an existing lightpath can be performed by comparing the current time with the reservation parameters of a lightpath, i.e., the start time and duration. In this model, the current time will be represented by the variable t. The function $K(u,t)$ is then defined such that if an existing lightpath u has begun utilizing its reservation, $K(u,t)=u$, otherwise $K(u,t)=\text{NULL}$. The function to check the status of an existing lightpath is formally defined by the following equation:

$$K(u,t)=\begin{cases} u & \text{if } t\geqslant S(u)\cap t\leqslant S(u)+D(u) \\ \text{NULL} & \text{otherwise} \end{cases}. \tag{4}$$

In Eq. (4) assume the functions $S(u)$ and $D(u)$ return the start time and duration of existing lightpath u, respectively.

Finally, it is now only a matter of labeling the links with the appropriate cost in order to compute the shortest path. However, for a link to be usable (i.e., have a cost less than ∞), those lightpaths that overlap in time and have not begun utilizing their reservation must be retunable. Therefore, we first define $A_{ij}^{\lambda v}$ to be the set of lightpaths overlapping in time with v on edge $(i^{\lambda},j^{\lambda})\in L^{\lambda}$. More formally, $A_{ij}^{\lambda v}$ is defined as follows:

$$A_{ij}^{\lambda v}=\{u\in U: Q(i^{\lambda},j^{\lambda},u,v)\neq\text{NULL}\}, \quad (i^{\lambda},j^{\lambda})\in L^{\lambda}, \quad \lambda\in\Lambda. \tag{5}$$

We can now label the cost of each link by simply evaluating the elements in $A_{ij}^{\lambda v}$. The cost function of link $(i^{\lambda},j^{\lambda})\in L^{\lambda}$ is defined by

$$C(i^{\lambda},j^{\lambda},v,t)=\begin{cases} 1 & \text{if } \forall u M(u)\neq\text{NULL}, \quad \forall u K(u,t)=\text{NULL}, u\in A_{ij}^{\lambda v}, \quad (i^{\lambda},j^{\lambda})\in L^{\lambda} \\ \infty & \text{otherwise} \end{cases}. \tag{6}$$

We show the mechanics of this algorithm in Fig. 1 by illustrating how the graph transformation and cost labeling will yield the minimum number of hops required to reserve a new connection request. With reference to Fig. 1, assume that the current topology represents the network configuration on a particular wavelength. Furthermore, assume that the set of lightpaths, i.e., $\{u_1,u_2,u_3,u_4,u_5,u_6\}$ all overlap in time with the new connection request. Furthermore, assume that every lightpath in the set

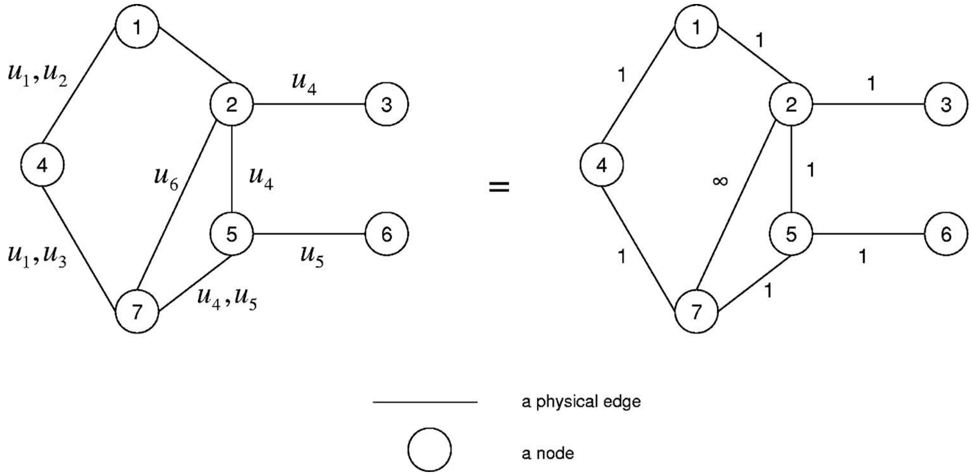

Fig. 1. Example illustrating the advance lightpath migration algorithm that minimizes the number of hops a new connection request traverses.

```
Λ := the set of wavelengths;
$L^\lambda$ := the set of links on wavelength $\lambda$;
$U$ := the set of reserved lightpaths;
$v$ := a new connection request;
$A$;   //The set of overlapping lightpaths on a particular link.
FOR each wavelength λ in Λ DO
     BEGIN for loop
     FOR each link $l$ in $L^\lambda$ DO
          BEGIN for loop
          $A := \emptyset$;
          FOR each lightpath $u$ in $U$ DO
               BEGIN for loop
               IF $u$ is using wavelength $\lambda$ AND link $l$ THEN
                    IF $u$ overlaps in time with $v$ THEN
                         IF $u$ is not in $A$ THEN add $u$ to $A$;
               END for loop
          $l.cost$ := 1;
          FOR each lighpath $u$ in $A$ DO
               BEGIN for loop
               IF $u$ has started the reservation OR $u$ is not retunable THEN
                    $l.cost := \infty$;
                    BREAK;
               END for loop
          END for loop
     END for loop
```

Fig. 2. Pseudocode for labeling the cost of each link on all wavelength subgraphs.

is both retunable and has not started to use its reservation with the exception of u_6, which is using edge (2, 7). If the new request demands a connection between nodes 1 and 7, the shortest path algorithm will find path 1-4-7 with cost 2. However, before the reservation can be made, lightpath reservations u_1, u_2, and u_3, must first be migrated to alternate wavelengths. The pseudocode for labeling the cost of each link on every wavelength subgraph follows this example and is presented in Fig. 2.

3.B. Minimize the Number of Migrated Lightpaths

In this subsection we wish to minimize the total number of lightpaths that need to be migrated in order to reserve a new connection request. In the graph transformation phase, we use a very similar method proposed in [13], where we add crossover edges along the route of a retunable lightpath, i.e., $[P(u,j)^\lambda, P(u,k)^\lambda]$: $H(u)+1 \geqslant k > j \geqslant 1, k-j \geqslant 2$. The sole reason for using crossover edges is so that while computing the shortest path, retunable lightpaths that traverse a series of links will not be counted more than once (this concept will be clarified later in an illustration of the algorithm). To create the graph with crossover edges, we first acquire the graph $\cup_{\lambda \in \Lambda} G(N^\lambda, L^\lambda)$ in the same way as we did in Subsection 3.A and then add crossover edges along the route of lightpaths that both overlap in time with v and are retunable, thereby creating a new graph $\cup_{\lambda \in \Lambda} G(V^\lambda, E^\lambda)$ where

$$V^\lambda = N^\lambda, \quad \lambda \in \Lambda, \tag{7}$$

$$E^\lambda = L^\lambda \cup X^\lambda, \quad \lambda \in \Lambda, \tag{8}$$

$$X^{\lambda}=\{[P(u,j)^{\lambda},P(u,k)^{\lambda}]:\ u\in U,\ H(u)+1\geqslant k>j\geqslant 1,k-j\geqslant 2\},\quad \lambda\in\Lambda. \tag{9}$$

The cost labeling phase for this objective is slightly more complicated since it is our intent to minimize the number of lightpaths which need to be retuned. It therefore follows that the cost of either a crossover edge or a physical edge be determined by the total number of retunable lightpaths traversing it. It should be noted again that a retunable lightpath is such that it overlaps in time with a new connection request v as well as can be migrated to a candidate wavelength. Furthermore, all overlapping lightpaths that traverse either a crossover edge or a physical link must be retunable and have not begun to use their reservation if the cost of the link is to be less than ∞.

For this objective, the cost of a physical edge is calculated by the function $C(i^{\lambda},j^{\lambda},v,t)$ where $(i^{\lambda},j^{\lambda})\in L^{\lambda}$, v is a new connection request, and t is the current time. The cost of a link will be equal to the sum of all elements in the set $A_{ij}^{\lambda v}$ if every overlapping lightpath on the link is retunable and is not currently using the reservation, or $C(i^{\lambda},j^{\lambda},v,t)=\varepsilon$ if there are no lightpaths on link $(i^{\lambda},j^{\lambda})\in L^{\lambda}$ that are overlapping in time with v, where ε is a very small positive value (e.g., 0.1) such that it is less than the length of the longest path, else $C(i^{\lambda},j^{\lambda},v,t)=\infty$ if at least one of the overlapping lightpaths on link $(i^{\lambda},j^{\lambda})\in L^{\lambda}$ is not retunable or has begun to use the reservation. Thus the cost of physical edge $(i^{\lambda},j^{\lambda})\in L^{\lambda}$ for the new connection request v is given by

$$C(i^{\lambda},j^{\lambda},v,t)$$

$$=\begin{cases}\displaystyle\sum_{u\in A_{ij}^{\lambda v}} u & \text{if } \forall\ uM(u)\neq \text{NULL},\ \forall\ uK(u,t)=\text{NULL},\ u\in A_{ij}^{\lambda v},\ (i^{\lambda},j^{\lambda})\in L^{\lambda},\ A_{ij}^{\lambda v}\neq\text{NULL}\\ \varepsilon & \text{if } A_{ij}^{\lambda v}=\text{NULL},\ (i^{\lambda},j^{\lambda})\in L^{\lambda}\\ \infty & \text{otherwise}\end{cases}. \tag{10}$$

To calculate the cost of a crossover edge, we must first identify the existing lightpaths that overlap in time with connection request v. Therefore, we define the set $B_{ik}^{\lambda v}$ that contains those lightpaths that overlap in time with the new connection request v on crossover edge $(i^{\lambda},k^{\lambda})\in X^{\lambda}$. More formally, $B_{ik}^{\lambda v}$ is defined as follows:

$$B_{ik}^{\lambda v}=\{u\in U{:}Q(N(i^{\lambda},k^{\lambda},j)^{\lambda},N(i^{\lambda},k^{\lambda},j+1)^{\lambda},u,v)\neq\text{NULL},\ j=1,\ldots,W(i^{\lambda},k^{\lambda})\},$$

$$(i^{\lambda},k^{\lambda})\in X^{\lambda},\quad \lambda\in\Lambda. \tag{11}$$

It should be noted that in the definition of $B_{ik}^{\lambda v}$, the functions $W(i^{\lambda},k^{\lambda})$ and $N(i^{\lambda},k^{\lambda},j)$ denote the total number of hops and the jth node along the route of crossover edge $(i^{\lambda},k^{\lambda})\in X^{\lambda}$, respectively.

The cost of crossover edge $(i^{\lambda},k^{\lambda})\in X^{\lambda}$ can now be calculated by $\hat{C}(i^{\lambda},k^{\lambda},v,t)$ where the cost of a crossover is equal to the sum of all elements in the set $B_{ik}^{\lambda v}$ if every overlapping lightpath on the link is both retunable and has not begun to use the reservation, and $\hat{C}(i^{\lambda},k^{\lambda},v,t)=\infty$ if at least one of the overlapping lightpaths on the crossover is either not retunable or has begun to use the reservation. Thus the cost of crossover edge $(i^{\lambda},k^{\lambda})\in X^{\lambda}$ is

$$\hat{C}(i^{\lambda},k^{\lambda},v,t)$$

$$=\begin{cases}\displaystyle\sum_{u\in B_{ik}^{\lambda v}} u & \text{if } \forall\ uM(u)\neq \text{NULL},\ \forall\ uK(u,t)=\text{NULL},\ u\in B_{ik}^{\lambda v},\ (i^{\lambda},k^{\lambda})\in X^{\lambda},\ B_{ik}^{\lambda v}\neq\text{NULL}\\ \infty & \text{otherwise}\end{cases}. \tag{12}$$

An example of this algorithm is shown in Fig. 3. If a new request specifies a connection between nodes 1 and 7, and again all existing lightpaths on the current subgraph both overlap in time with the new request are both retunable and have not begun to

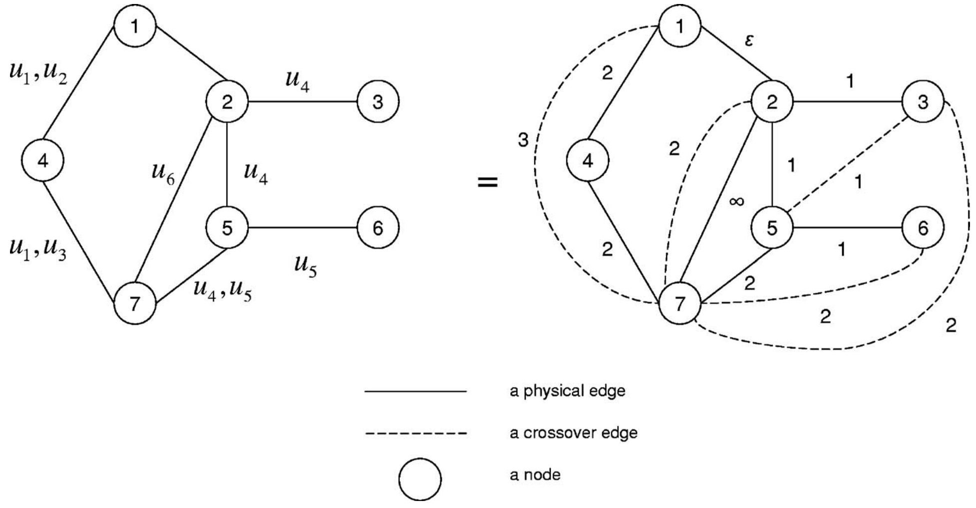

Fig. 3. Example illustrating the advance lightpath migration algorithm that minimizes the number of migrated lightpaths.

```
Λ := the set of wavelengths;
$L^\lambda$ := the set of links on wavelength $\lambda$;
$U$ := the set of reserved lightpaths;
$v$ := a new connection request;
ε := a value less than the length of the longest path;
$A$;   //The set of overlapping lightpaths on a particular link.
FOR each wavelength λ in Λ DO
    BEGIN for loop
    FOR each link $l$ in $L^\lambda$ DO
        BEGIN for loop
        $A := \emptyset$;
        FOR each lightpath $u$ in $U$ DO
            BEGIN for loop
            IF $u$ is using wavelength $\lambda$ AND link $l$ THEN
                IF $u$ overlaps in time with $v$ THEN
                    IF $u$ is not in $A$ THEN
                        add $u$ to $A$;
            END for loop
        IF $A$  is not empty THEN
            FOR each lighpath $u$ in $A$ DO
                BEGIN for loop
                IF $u$ has started the reservation OR $u$ is not retunable THEN
                    $l.cost := \infty$;
                    BREAK;
                ELSE $x.cost$++;
                END for loop
        ELSE $l.cost$ := ε;
        END for loop
    END for loop
```

Fig. 4. Pseudocode to label the cost of each link when minimizing the number of migrated lightpaths.

utilize their reservations with expection of lightpath, u_6 the shortest path algorithm should find path 1-2-7 (equivalent to 1-2-5-7) with a cost of 2.1, where $\varepsilon = 0.1$. It should be noted in the figure that the crossover edge (2,7) refers to the physical links (2,5) and (5,7), respectively. Here, we see that the algorithm chooses the route that minimizes the number of lightpaths which need to be migrated to accommodate a new connection request. Presented in Figs. 4 and 5 is the pseudocode used to label the cost of each physical link and crossover link when employing the minimum number of migrated lightpaths algorithm, respectively.

3.C. Complexity

Both advance lightpath migration algorithms can be broken into two phases. The first phase is bounded by first calculating the cost of each link. This process requires the

```
$\Lambda$ := the set of wavelengths;
$L^\lambda$ := the set of links on wavelength $\lambda$;
$U$ := the set of reserved lightpaths;
$X^\lambda$ := the set of crossover links on wavelength $\lambda$;
$v$ := a new connection request;
$B$;   //The set of overlapping lightpaths on a particular crossover link.
FOR each wavelength $\lambda$ in $\Lambda$ DO
    BEGIN for loop
    FOR each crossover link $x$ in $X^\lambda$ DO
        BEGIN for loop
        $B := \emptyset$;
        FOR each link $l$ along the route of the crossover link $x$ DO
            BEGIN for loop
            FOR each lightpath $u$ in $U$ DO
                BEGIN for loop
                IF $u$ is using wavelength $\lambda$ AND link $l$ THEN
                    IF $u$ overlaps in time with $v$ THEN
                        IF $u$ is not in $B$ THEN add $u$ to $B$;
                END for loop
            END for loop
        FOR each lighpath $u$ in $B$ DO
            BEGIN for loop
            IF $u$ has started the reservation OR is not retunable THEN
                $x.cost := \infty$;
                BREAK;
            ELSE $x.cost$++;
            END for loop
        END for loop
    END for loop
```

Fig. 5. Pseudocode to label the cost of each crossover link when minimizing the number of migrated lightpaths.

algorithm to build the set of lightpaths that overlap in time with the new request, $O(|\Lambda\|L\|U|)$, and then evaluate whether each of those lightpaths can be retuned to alternate wavelengths, $O(|\Lambda|^2|L|^2|U|)$. If we are using the migration algorithm which minimizes the total number of migrated lightpaths, we will also need to build the set of crossover links, $O(|U\|N|^2)$, and calculate the cost of each one, $O(|\Lambda\|L|^2|U|^2|N|^2)$. In the second phase, we simply perform Dijkstra's shortest path algorithm on each wavelength subgraph, which again has a time complexity of $O(|\Lambda\|N|^2)$.

4. Performance Evaluation

In this section, we evaluate the performance of the proposed rerouting algorithms for an advance lightpath reservation WDM network.

4.A. Simulation Model

The network topology used in our simulations is the 16-node nationwide network NSFNET (Fig. 6). For our simulation purposes, it is assumed that each of the 25 links in NSFNET is bidirectional and is equipped with 16 wavelengths. Our traffic model generates new connection requests for a period of 24 h, where each new request is uniformly distributed among all source and destination pairs. New requests arrive according to a Poisson process at an arrival rate of λ. Similarly, each new request is accompanied by a holding time that follows an exponential process where the average holding time $1/\mu = 30$ min. To the best of our knowledge there is no distribution curve that can effectively represent advance reservation traffic; therefore the start time of a request is uniformly distributed between a predefined window size and the calculated arrival time. In our simulations the window size is assumed to be a length of 2 h. To study the network and algorithms under different loads the average arrival rate of a connection request is varied as a simulation parameter. The load in the network is measured in Erlangs, and is calculated by multiplying the average arrival rate with the average holding time. All algorithms were implemented using Microsoft Visual Studio.NET and written in C++. All tests were performed on a desktop PC with a Pentium 4 3 GHz processor and 2 Gbytes RAM running Microsoft Windows XP Professional.

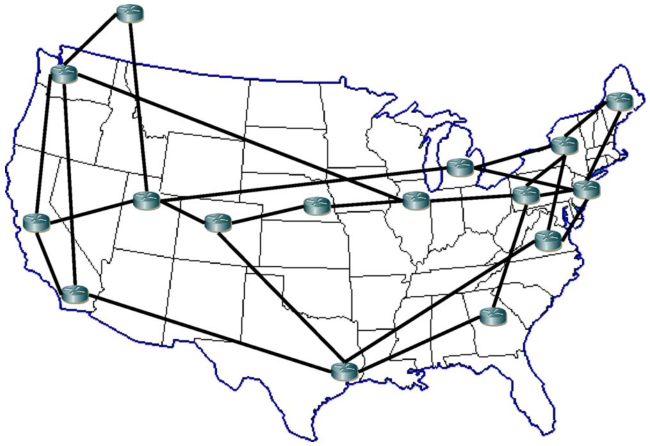

Fig. 6. Nationwide backbone network topology NSFNET.

4.B. Blocking Probability

Blocking probability refers to the probability of a connection request not being reserved due to the unavailability of resources. In an advance reservation WDM network, if a connection request cannot find a continuous wavelength route between the specified start time and end time, it will be blocked. However, to reduce the blocking probability, we can migrate reserved lightpaths to candidate wavelengths along the same path in attempt to free enough resources so that a new connection request can be reserved. Figure 7 shows the blocking probability versus the traffic load for continuous routing (i.e., no rerouting) and the two different rerouting algorithms. By employing lightpath migration, the advance reservation network of Fig. 6 experiences the greatest difference in blocking probability at a load of 5.5 Erlangs with a 23% improvement over continuous routing.

4.C. Average Number of Hops Traversed per Rerouting

Some WDM systems may demand that new connection requests traverse the fewest number of hops possible. This type of service can benefit distributed systems where

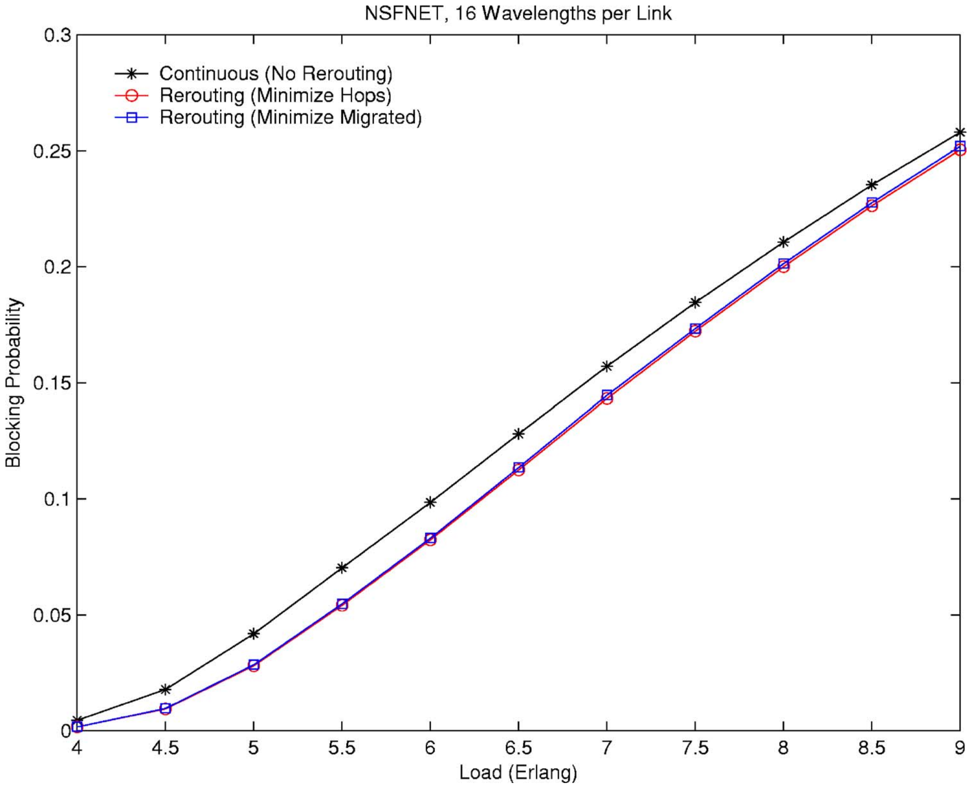

Fig. 7. Blocking probability versus load.

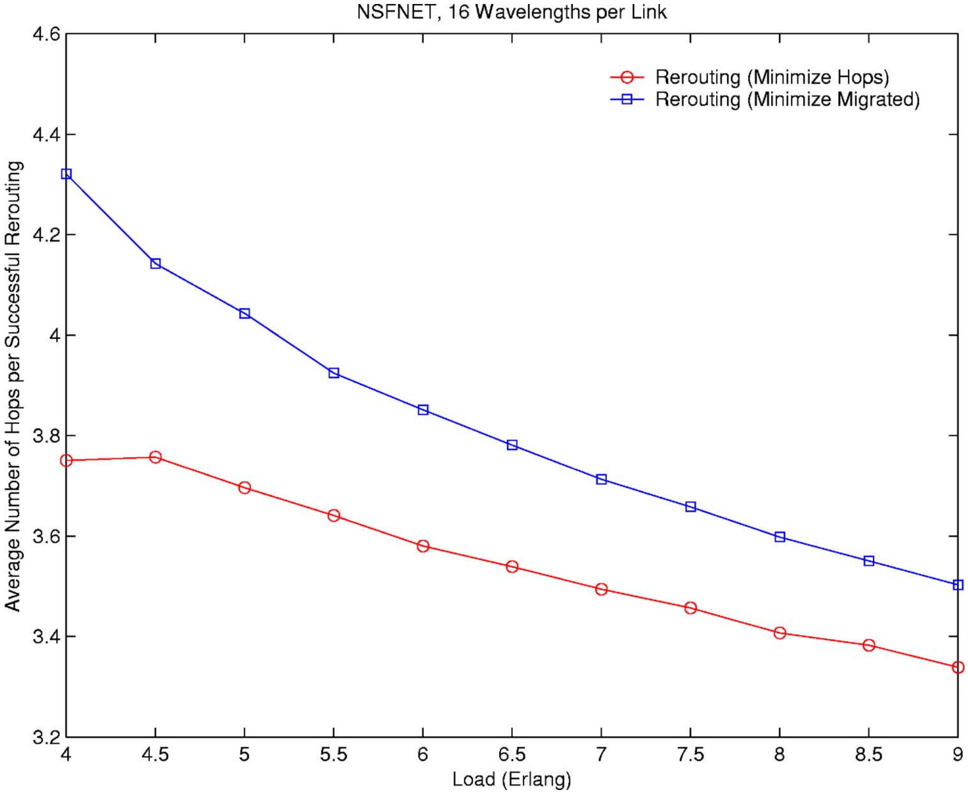

Fig. 8. Average number of hops per rerouting versus load.

connections are established through control messages as well as alleviate computational complexity. For example, it should be less likely that a reservation message encounters a contention while establishing a new connection if there are a low number of hops. Figure 8 shows the average number of hops a new connection request traverses after successful rerouting. It can be seen in the plot that as the load increases, the average number of hops new connections traverse tends to reduce. This shows that only shorter-hop requests will be rerouted as the network becomes more congested. However, the most important observation in this experiment is that we have shown that on average our algorithm that minimizes the number of hops of a new request (i.e., after lightpath migration) performs the best. And therefore, we can confirm its accuracy through experimentation.

4.D. Average Number of Migrated Lightpaths per Rerouting

This subsection evaluates the performance of each algorithm in terms of the average number of migrated lightpaths per successful rerouted request. The main idea behind this metric is to characterize the time required by a centralized controller to make the appropriate configuration changes during lightpath reservation and migration. In Fig. 9 we see that the algorithm that minimizes the number of migrated lightpaths is empirically validated since it consistently produces the best results. It can be realized from the graph that when the network is lightly loaded, fewer numbers of existing lightpaths require migration to accommodate a new request because more resources are available. Therefore, as the load begins to increase, it becomes more difficult to find routes without having to perform lightpath migration. However, after the network becomes congested, only lightpath requests that traverse a few hops will be accepted. Therefore, the number of migrated lightpaths will begin to decrease.

5. Conclusions and Future Work

In this paper we introduced the advance reservation dynamic lightpath establishment problem. For this problem, we proposed a new model that reserves lightpaths in advance for a centralized WDM network with dynamic traffic. This model is advantageous because existing lightpaths can be migrated to candidate wavelengths to allow

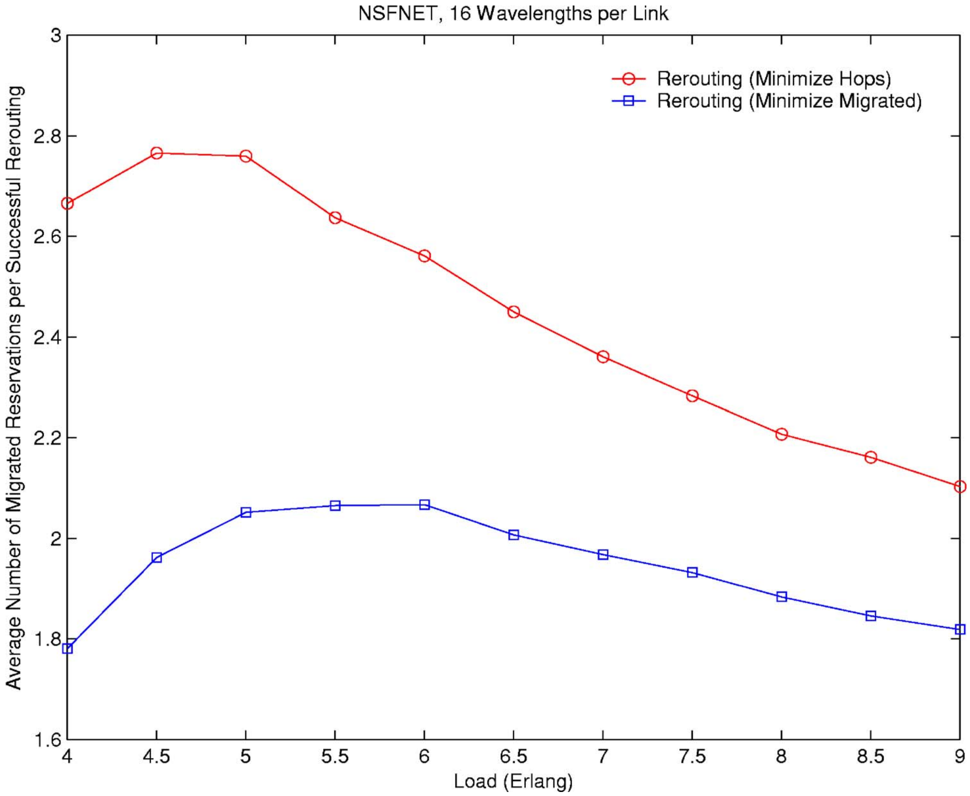

Fig. 9. Average number of migrated lightpaths per rerouted request versus load.

new requests to be reserved. Through our simulation results, we can conclude that the lighpath migration scheme will significantly reduce system blocking probability.

Moreover, we discussed two ways of attempting to optimize the efficiency of our advance lightpath migration model by suggesting the following objective functions: (1) minimize the number of hops a new request traverses after migration, or (2) minimize the number of migrated lightpath. Finally, we validated each objective through simulation and empirical studies.

In this work, it was assumed that advance requests are reserved for a specific start time, or blocked if resources are unavailable. Since advance reservation applications may have flexible start times (e.g., bulk file transfers), we propose a scheduling problem for advance lightpath reservation: *minimize the tardiness of a new request*. In this problem, requests arrive dynamically, and the current system configuration is updated such that all current reservations must retain their reserved start time, but change paths and/or wavelengths such that the tardiness of the new advance connection request is minimized. By using our new advance lightpath migration algorithm, the objective to minimize the tardiness of a new request can easily be achieved by continuously updating the requested start time to the end time of the reservation with the soonest end time that also overlaps in time with the new request. It would therefore be the intention of the network to provide the next earliest start time for an advance lightpath connection request.

References

1. J. Taylor, “Defining e-science,” *National e-Science Centre*, retrieved May 25, 2006, from http://www.nesc. ac. uk/nesc/define. html.
2. “NIH Provides $32.8 million to enhance biomedical informatics research network,” *National Institutes of Health*, Press Release (December 8, 2004), Retrieved May 24, 2006, from http://www.nih. gov/news/pr/dec2004/ncrr-08.htm.
3. “LHC computing centers join forces for global grid challenge,” *CERN DSU-Communication*, Press Release (April 25, 2005), retrieved May 24, 2006, from http://info.web.cern.ch/Press/ Press Releases/Releases2005/PR06.05E. html.
4. C. Kesselman, I. Foster, and T. Prudhomme, *The Grid: Blueprint for a New Computing Infrastructure* (Elsevier, 2004), pp. 81–93.
5. D. Kuo and M. Mckeown, “Advance reservation and co-allocation protocol for grid

computing," in *Proceedings of the First International Conference on e-Science and Grid Computing* (IEEE, 2005) pp. 164–171.

6. K. Czajkowski, I. Foster, and C. Kesselman, "Resource co-allocation in computational grids," in *Proceedings of the Eighth International Symposium on High Performance Distributed Computing* (IEEE, 1999) pp. 219–228.
7. J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, and R. Douville, "A routing and wavelength assignment of scheduled lightpath demands," IEEE J. Sel. Areas Commun. **21**, 1231–1240, (2003).
8. J. Kuri, N. Puech, and M. Gagnaire, "Diverse routing of scheduled lightpath demands in an optical transport network," in *Design of Reliable Communication Networks (DRCN)* (IEEE, 2003), pp. 69–76.
9. M. Veeraraghavan, X. Zheng, W. Feng, H. Lee, E. Chong, and H. Li, "Scheduling and transport for file transfers on high-speed optical circuits," J. Grid Comput. **1**, 395–405 (2003).
10. J. Zheng and H. T. Mouftah, "Supporting advance reservations in wavelength-routed WDM networks," in *Proceedings of 10th International Conference on Computer Communications and Networks* (IEEE, 2001), pp. 594–597.
11. J. Zheng and H. T. Mouftah, "Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks," in *IEEE International Conference on Communications (ICC 2002)* (IEEE, 2002), Vol. 5, pp. 2722–2726.
12. J. Zheng and H. T. Mouftah, "A framework for supporting advance reservation service in GMPLS-based WDM networks," in *Proceedings of 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 2003)* (IEEE, 2003), Vol. 1, pp. 486–489.
13. K. C. Lee and V. O. K. Li, "A wavelength rerouting algorithm in wide-area all-optical networks," J. Lightwave Technol. **14**, 1218–1229 (1996).
14. G. Mohan and C. Siva Ram Murthy, "A time optimal wavelength rerouting for dynamic traffic in WDM networks," J. Lightwave Technol. **17**, 406–417 (1999).