Scheduling of Periodic Connections with Flexibility[1]

Wei Su[a], Galen Sasaki[b,*], Ching-Fong Su[c], and Ashok Balasubramanian[b]

[a]*Leadtek Research Inc., 46732 Lakeview Blvd., Fremont, CA 94538, USA*
[b]*Dept. of Electrical Engg., University of Hawaii at Manoa, 2540 Dole St., Honolulu, HI 96822, USA*
[c]*Fujitsu Laboratories of America, 1240 E. Arques Ave. #345, Sunnyvale, CA 94085, USA*

Abstract

Applications that require periodic optical connectivity may be serviced by scheduled lightpaths. Many applications have flexibility about when exactly the scheduled lightpaths should occur. Lightpath service providers can exploit the flexibility to schedule the connections to better utilization of resources, and presumably lower prices for clients. To find the relationship between network resources and time flexibility, a simple WDM link and a leaky bucket traffic model are considered. The network resources, and in particular the required number of wavelengths W, is shown to be a function of the traffic parameters and the time flexibility. For random traffic, simulations are used to measure the sensitivity of W with time flexibility.

Keywords: WDM; scheduled lightpaths; periodic connectivity; time flexibility; performance evaluation.

1 Introduction

Scheduled lightpaths [1] [2] are required to support applications requiring optical connections with known set-up and tear-down times. For example, an IP service provider may build its network using lightpaths as IP links leased from a lightpath service provider. The IP network has a baseline topology that provides guaranteed network capacity at all times. The links are realized by static lightpaths. However, during peak traffic hours (e.g., working hours of 8am to 5pm), the IP network needs additional links. The additional links can be realized by *periodic lightpaths* [3] that are scheduled from 8am to 5pm daily. Another example is an office that requires a daily Gigabit Ethernet connection to a remote data storage facility to back-up data, say starting at 12 midnight for 30 minutes.

An alternative to scheduled periodic lightpaths are *dynamic* lightpaths [4], where users can quickly set up and tear down lightpaths with their lightpath service providers. However, dynamic service is less reliable. For example, consider the earlier scenario where an office backs-up its data at a remote storage facility. The office must set-up a dynamic lightpath per day at midnight from a service provider. However, if, for some particular day, the service provider experiences an unusually high load then the office

[1] An earlier version of this paper appeared in [3]. To appear in Journal of Optical Switching and Networking.

[*] Corresponding author. Tel.: +1 808 956 7586; fax: +1 808 956 3427.
Email addresses: weisu@ieee.org (W. Su), sasaki@spectra.eng.hawaii.edu (G. Sasaki), chingfong.su@us.fujitsu.com (C. –F. Su), balasubr@hawaii.edu (A. Balasubramanian).

may be denied service. On the other hand, a periodic lightpath service [3], once set up, provides a *guaranteed* lightpath starting at the same time every day.

The costs for a scheduled lightpath may be based on usage-based pricing by the service provider, such as daily duration, bit rate and other requirements of the customer. Also it may depend on Time-of-Day (ToD) pricing [5] [6] [7] for the lightpath service, e.g., a start time during peak traffic hours will be more expensive than light traffic hours. In this paper, we consider another feature of scheduled lightpaths that has not been analyzed in the current literature. We refer to this as *time flexibility*. To describe this feature, let us recall the storage back-up example. Let the office require 30 minutes of lightpath service per day over a service period of a year. But now suppose the service does not have to start at 12 midnight. However it must start at some fixed time every day because the back-ups must be done *regularly*. Any fixed time will do as long as no one is working in the office, e.g., between 12 midnight and 4 am. Thus, the office is flexible with respect to the start time of the lightpath service. When the office applies for a periodic scheduled lightpath from a service provider, it can include its flexibility. Then the provider can choose a start time that will efficiently utilize its resources. For example, the provider may choose 3 am for the start time because then it is the least busy period. Then the provider notifies the office of its service including the start time. Subsequently the office will have a lightpath for 30 minutes every day at 3 am for a period of a year. Note that flexibility in start time during set-up of periodic network services will lead to better utilization of resources for the service provider, and presumably lower prices on lightpath services for the customer.

In this paper, we will consider scheduled lightpaths that are periodic and have time flexibility. We investigate how the flexibility of service starting time may improve the utilization of network resources, and may have an effect on the price of services. Network cost and its dependence on time flexibility will be analyzed. In this paper, we take the network cost to be the number of wavelengths. Also, we consider the problem of scheduling these lightpaths, and present efficient algorithms to solve the problem.

We focus on a simple scenario. There is a single fiber-link with W wavelengths. Though our discussion will focus on a single link, it also applies to unidirectional WDM ring networks, where scheduled lightpaths are full duplex.

We assume time is organized into time slots, that are numbered 0, 1, ..., T-1, where T denotes the number of time slots in a day. For example, we can assume time slots 0 and T/2 are, respectively, 12 midnight and 12 noon. Throughout the paper, we will use the following terms for subsets of the interval $[0, T-1]$. An *ordinary interval* $[s,t]$ is just a subinterval of $[0, T-1]$, where $s \leq t$. If $s,t \in [0, T-1]$ and $s > t$ then $[s,t]$ is a *wrap around interval* which corresponds to the time slots $[t, t+1, ..., T-1, 0, 1, ... s]$. The size of an interval $[s,t]$ is denoted by $\|[s,t]\|$. A *scheduled lightpath* for the link is characterized by a triple (w, s, L), which means wavelength w is used for a *duration* of L time slots starting from slot s per day. Since the service repeats every T time slots, we assume without loss of generality that $L \leq T$. The service duration may wrap around from the end of a day to the beginning of the next. For example, the lightpath service $(w, T-2, 4)$ is for the time slots [*T*-2, *T*-1, 0, 1].

A *request* for lightpath service is a triple (a, b, L), where the service has duration *L*, and the first time slot of the service must be in the time interval $[a, b]$. The interval $[a, b]$ is referred to as the request's *start window*, and the value $\|[a, b]\| - 1$ is its *time flexibility*. The time flexibility ranges from 0 to $T-1$, where 0 corresponds to no flexibility, and $T-1$ corresponds to no restrictions. The parameters *a* and *b* are referred to, respectively, as the *earliest* and *latest start times* for the lightpath. A request (a, b, L) can be *assigned* to a lightpath service (w, s, L') if $L' = L$ and $s \in [a, b]$.

We assume that there is a batch of requests for lightpath service for the WDM link, which is empty at time 0. A *feasible assignment* of the requests to lightpath services is one where there is at most one lightpath service per wavelength at all times. An example feasible assignment is shown in Figure 1, when *W* = 2 and *T* = 8. The lightpath requests are (4,6,4), (3,3,2), (7,1,3), and (1,3,4). Note that the lightpath service for (4,6,4) wraps around. Also note that (3,3,2) has no flexibility and the other requests have time flexibilities of 2.

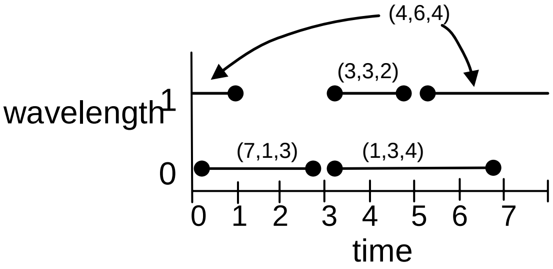


Figure 1. An assignment of four scheduled lightpaths.

An alternative to modeling traffic as a batch-of-requests is to model requests as arriving in some sequence over time. Then the requests are assigned periodic scheduled lightpaths, one at a time, as they arrive. The batch model better highlights improvements due to time flexibility because the optimization of assignments can be better coordinated over the entire batch. In this paper, we only consider the batch-of-requests traffic model.

We consider how time flexibility can lead to better use of resources, and in particular minimizing the number of wavelengths. In Section 2, a simple leaky bucket traffic model [8] is applied to model lightpath requests. The model captures the key characteristics of burstiness, peak-load, and average-load. We assume all lightpath requests have the same time flexibility. The traffic model leads to closed-form formulas that show a relationship between the number of wavelengths, the time flexibility, and the parameters of the traffic. The accuracy of these formulas is discussed. In Section 3, traffic with different time flexibilities is considered.

Simulations to measure how the number of wavelengths depends on time flexibility are presented in Section 4, where we also compare the simulation results with the formulas derived in Section 2. Heuristic assignment algorithms were applied because the problem of finding a feasible assignment that uses a minimum number of wavelengths is NP Complete. The simulation results indicate that there can be significant reductions in the number of wavelengths with a modest amount of time flexibility. Our final remarks are given in Section 5.

Next, we discuss other related work. *Time division multiplexed* (TDM), WDM networks, as studied in [9], [10], and [11], focus on assigning time slots in a periodic frame and routing for constant-bit-rate, sub-wavelength connections. The concept of *scheduled lightpaths* is presented in [1] and [2] but without time flexibility. The problem of scheduling on a single WDM link with no flexibility is related to the *circular interval graph coloring* problem [14]. In [14], upper bounds were given on the number of colors needed for the problem.

Advance reservations in [12] and [13] considered earliest and latest start times in connection requests and the complexity of selecting paths in a network. Our heuristics use techniques based on *variable and value ordering* that were used in [15] to solve the job shop scheduling problem as a *constraint satisfaction problem* (CSP). The work in [16] and [17] proposed traffic models that consider both earliest and latest start times of connection requests, but there is no detailed study of the relationship between flexibility in start times and the number of required wavelengths. In addition, we are unaware of any earlier work that applies the leaky bucket traffic model in the context of scheduled lightpaths as we do in this paper. Nor are we aware of any earlier work that has closed-form expressions relating the number of wavelengths, time flexibility, and traffic parameters.

2 Wavelength Efficiency for Single Time Flexibility

In Subsection 2.1, we describe the traffic model, which is essentially the same traffic model introduced by Cruz [8] for packet traffic. In Subsection 2.2, we present preliminary lemmas, and in Subsection 2.3, we have formulas for the required number of wavelengths and a discussion of their accuracy.

2.1 Traffic Model

We view the batch of lightpath requests as *virtual packets* as follows. A lightpath request (a, b, L) is also referred to as a *virtual packet* in a *virtual packet-switched, queueing system*, where the wavelengths are transmission links. The virtual packet “arrives” at time *a*, and has “transmission duration” *L*. The earliest and latest it can start transmission is, respectively, *a* and *b*. Thus, its “queueing delay” is at most $\|[a,b]\| - 1$, i.e., the time flexibility of the lightpath request.

For a lightpath request (a, b, L), the lightpath service can occur anywhere in the interval $[a, b+L-1]$. We refer to the length of this interval as the *spread* for the request. Note that the spread is the sum of the time flexibility and *L*. The spread is the "total delay" of the virtual packet.

For each time *t*, let $A(t)$ denote the sum of transmission durations of the virtual packets that arrive at time *t*. We refer to this as the *work* that arrives at time *t*. Let *W* be the number of wavelengths required to schedule the lightpath requests.

We will characterize a batch of lightpath requests with the parameters (σ, ρ, π), where ρ is a measure of the average traffic load, σ is a measure of the traffic "burstiness", and π is the peak load. We say that the lightpath requests are (σ, ρ, π) *constrained* if for all time intervals [*x*, *y*],

$$\sum_{t\in[x,y]} A(t) \le \min\{\sigma + \rho|[x,y]|, \pi|[x,y]|\}.$$

In addition, we assume that $\sum_{t\in[0,T-1]} A(t) \le \rho T$ and $\rho \le W$ so that the average requested traffic load will be within the capacity of the fiber-link. Without loss of generality, we also assume that $\pi > \rho$.

2.2. Preliminary Results

We present two preliminary results in Lemmas 2.1 and 2.2. Lemma 2.1 shows that we can split the traffic approximately evenly among the wavelengths. Lemma 2.2 describes how traffic can be assigned time slots on a given wavelength.

Lemma 2.1 is for the following simple load-balancing algorithm. The algorithm attempts to distribute the virtual packets to the wavelengths so that each wavelength gets a fraction 1/*W* of the work at all times, i.e., each wavelength gets about $A(t)/W$ of the work. The algorithm keeps track of virtual packets (and their work) assigned to each wavelength, where initially no virtual packets are assigned.

To be more specific, the algorithm considers virtual packets in the order they arrive starting from time 0. For each wavelength *k*, let $A_k(t)$ denote the work of virtual packets that are assigned to the wavelength and arrive at time *t*. The virtual packets that arrive at time *t* are considered one at a time in some arbitrary order and are assigned to the wavelengths with the least work, i.e., assigned to wavelength *k* with the smallest $\sum_{n\in[0,t]} A_k(n)$.

The load balancing has the following properties. Let L_{max} denote the longest service duration required in the batch of requests. The load balancing algorithm ensures that between any pair of wavelengths *j* and *k*, $\left|\sum_{n\in[0,t]} A_j(n) - \sum_{n\in[0,t]} A_k(n)\right| \le L_{\max}$ for all *t*.

Also note that $\sum_{n\in[0,t]} A(n)/W$ is the average of $\sum_{n\in[0,t]} A_k(n)$ over all wavelengths *k*. Thus, for any wavelength *k* and time *t*,

$$\left|\sum_{n\in[0,t]} A_k(n) - \sum_{n\in[0,t]} A(n)/W\right| \le L_{\max} \,. \tag{2.1}$$

Now consider an arbitrary ordinary interval [*s*,*t*]. Note that $\left|\sum_{n\in[s,t]} \left(A_k(n) - A(n)/W\right)\right|$ $= \left|\sum_{n\in[0,t]} \left(A_k(n) - A(n)/W\right) - \sum_{n\in[0,s-1]} \left(A_k(n) - A(n)/W\right)\right| \le \left|\sum_{n\in[0,t]} \left(A_k(n) - A(n)/W\right)\right|$ $+ \left|\sum_{n\in[0,s-1]} \left(A_k(n) - A(n)/W\right)\right|$. By applying Inequality (2.1) to the last expression, we have the following. For any wavelength *k* and ordinary interval [*s*,*t*],

$$\left|\sum_{n\in[s,t]} A_k(n) - \sum_{n\in[s,t]} A(n)/W\right| \le 2L_{\max} \,. \tag{2.2}$$

Next consider a wrap around interval $[s,t]$. It is composed of two ordinary intervals $[s, T-1]$ and $[0,t]$. We can apply (2.1) and (2.2), to get the following. For any wavelength *k* and wrap around interval $[s,t]$,

$$\left|\sum_{n\in[s,t]} A_k(n) - \sum_{n\in[s,t]} A(n)/W\right| \le 3L_{\max} \,. \tag{2.3}$$

Thus, inequality (2.3) is true for any interval $[s,t]$. This implies the following lemma.

Lemma 2.1. Consider a batch of lightpath requests and its virtual packet model. Let *A*(*t*) denote the work of the virtual packets that arrive at time *t*. Let L_{max} denote the length of the longest request. There is an algorithm that can split the batch of requests into *W* batches with the following properties. For *k* = 1, 2, …, *W*, let $A_k(t)$ denote the work of the virtual packets that arrive at time *t* for batch *k*. Then, for every interval [*s*,*t*],

$$\left|\sum_{n\in[s,t]} A_k(n) - \sum_{n\in[s,t]} A(n)/W\right| \le 3L_{\max} \,.$$

We now turn our attention to Lemma 2.2. Consider an arbitrary wavelength *k* and its batch of requests. We will apply the following method to assign time slots to virtual packets. We pick a starting time slot *s*. We model the wavelength as a virtual packet-switched, queuing system which has some work conserving service policy, e.g., *first come first serve* or *earliest deadline first*. Just before the initial time *s*, the system is empty. If *s* = 0, then the system runs until time slot *T*-1. If *s* > 0, then the system runs from time slot *s* to time slot *s*-1, where time wraps around from *T*-1 to 0. Note that the virtual packets in queue at time slot *T*-1 are carried over to time slot 0 when time wraps around. The virtual packets, for wavelength *k*, are served according to the service policy.

The method to assign time slots to virtual packets requires both a starting time slot s and a service policy. At the final time slot s-1, there may be virtual packets still in queue, or the packet being serviced may require additional time slots. Then we say that the starting time slot s leads to *work overlap* for the service policy. The next lemma shows that under reasonable traffic conditions, there is a starting time slot s that leads to no work overlap.

Lemma 2.2. Consider a wavelength k and the lightpath requests that are assigned to it. Suppose $\sum_{t=0}^{T-1} A_k(t) \leq T$. We model the system using the virtual packet-switched, queueing system model. Then, for any work conserving service policy, there is a starting time slot s that does not lead to work overlap.

Proof. We call a time interval [x,y] a *single-backlog* interval if the following two properties are true. First, the starting time slot x has arrivals of virtual packets. Second, suppose the system is empty except for the virtual packets that arrived in [x,y], and the service policy is used on the virtual packets starting from time slot x. This creates a *single busy period* [x,z]. We say that the busy period *overflows* if $\|[x,z]\| > \|[x,y]\|$, i.e., it goes beyond time slot y.

Now, for each time slot x that has virtual packet arrivals, we create single-backlog intervals [x,x]. If a single-backlog interval has a busy period that overflows into the next single-backlog interval then merge the two intervals and apply the service policy starting from the beginning of the new interval. Note that the new interval is itself a single-backlog interval. Also note that we cannot have a situation where there is just one single-backlog interval (i.e., it covers all T time slots) that overflows into itself. Otherwise, its busy period is over T time slots, which contradicts the constraint $\sum_{t=0}^{T-1} A_k(t) \leq T$. When there are no possible mergings, there will be only single-backlog intervals that do not overflow.

The beginning of one of the single-backlog intervals can be a starting time slot s. Since none of the single-backlog intervals overflow into the next single-backlog interval, such a starting time slot does not lead to work overlap. Hence, the lemma is true. ■

Next we show a stability condition that leads to $\sum_{t=0}^{T-1} A_k(t) \leq T$ for all wavelengths k.

Lemma 2.3. Suppose there is a batch of lightpath requests that are (σ,ρ,π) constrained. Let the durations of the requests be at most $L_{\max}$. Suppose $\pi \geq W \geq \rho$,

$$W \geq \rho\left(1+\frac{3L_{\max}}{T-3L_{\max}}\right) \tag{2.4}$$

Then, for all wavelengths k, $\sum_{t=0}^{T-1} A_k(t) \leq T$, and there is no work overlap. The queue system is then called *stable*.

Proof. Lemma 2.1 implies a splitting of lightpath requests to the W wavelengths. Consider an arbitrary wavelength k, and let $A_k(t)$ denote the work of virtual packets that are assigned to the wavelength and arrive at time t. The lemma implies, for any interval $[s,t]$, $\sum_{n\in[s,t]} A_k(n) \le 3L_{\max} + \sum_{n\in[s,t]} A(n)/W$. Since the lightpath requests are (σ,ρ,π) constrained,

$$\sum_{n\in[s,t]} A_k(n) \le 3L_{\max} + \min\{\sigma' + \rho' |[s,t]|, \pi' |[s,t]|\}, \tag{2.5}$$

where $\sigma' = \frac{\sigma}{W}$, $\rho' = \frac{\rho}{W}$, and $\pi' = \frac{\pi}{W}$.

In addition, the lemma implies $\sum_{n\in[0,T-1]} A_k(n) \le 3L_{\max} + \sum_{n\in[0,T-1]} A(n)/W$ $\le 3L_{\max} + \frac{\rho T}{W}$. Inequality (2.4) implies $\frac{\rho T}{W} \le T - 3L_{\max}$. Therefore, $\sum_{n\in[0,T-1]} A_k(n) \le T$, and Lemma 2.2 can be applied to wavelength k and its batch of requests. ■

Condition $d_{\min} > 3L_{\max} + 1$ and Inequality (2.4) are due partly to the imperfect load balancing of Lemma 2.1. Inequality (2.4) is also due to the fact that the lightpaths can have durations that make them impossible to pack into wavelengths efficiently. To illustrate this, consider the following batch of requests. Let n, k, and L be arbitrary integers. Let $T = nL - 1$, all lightpaths have duration L, and all time slots have k arriving requests. Let $\rho = kL$, $\sigma = 0$, and $\pi = \infty$. Note that at most $n-1$ lightpaths can fit into a wavelength, leaving at least $L-1$ unused time slots. Since there are kT lightpaths, the number of wavelengths must be at least $\frac{kT}{n-1} = \frac{kT}{(T+1)/L - 1} = \rho\left(1 + \frac{L-1}{T+1-L}\right)$. Note the similarity between the last term and Inequality (2.4).

The next corollary follows from the theorem and the fact that the spread of a lightpath request is the sum of its duration and time flexibility.

2.3. Wavelength Requirements

The following is the main result of the section, which presents the relationship between the wavelength requirement and the spreads of the requests. At the end of the subsection, we give an interpretation of the results.

Theorem 2.1. Suppose there is a batch of lightpath requests that are (σ,ρ,π) constrained. Let the durations of the requests be at most $L_{\max}$. Let the spreads of the requests be at least $d_{\min}$, and $d_{\min} > 3L_{\max} + 1$. Suppose $\pi \ge W \ge \rho$, and inequality (2.4) is held

and

$$W \geq \frac{\pi}{1+\left(d_{\min}-3L_{\max}-1\right)/\left(\frac{\sigma}{\pi-\rho}\right)} \tag{2.6}$$

Then there is a feasible assignment for the batch of lightpath requests, where all packets are scheduled within their spreads.

Proof. Given (2.4), Lemma 2.2 and 2.3 imply that for any work conserving service policy, there is a starting time slot *m* that leads to no work overlap. Therefore, we can treat the virtual packet-switched, queueing system as an ordinary packet switched system over the time interval [*m*, *m*-1]. In addition, the traffic is constrained by Inequality (2.5).

Consider first come first served as the service policy. Consider an arbitrary virtual packet *i* and let its arrival time be denoted by *t*. Let $g(t)$ denote the remaining amount of work at time slot *t*. This includes residual work from packets that arrive before time *t*, and work of packets that arrive at time *t*, including virtual packet *i*. Since the service policy is first come first served, packet *i* will complete service within an interval of length $g(t)$ beginning from time *t*. Note that if $g(t)$ is within the spread of packet *i* then the packet will be served within its time flexibility.

We now proceed to find an upper bound for $g(t)$. Packet *i* arrives in a busy period, and let *t'* denote the beginning of the busy period. Note that $g(t)$ is the difference between the amount of traffic that arrived during the busy period up until time *t*, and the amount of work done during the busy period before time *t*. Using (2.5), one can obtain

$$\begin{aligned} g(t) &= \sum_{n=t'}^{t} A_k(n) - \left(\left[t', t-1\right]\right) \\ &\leq 3L_{\max} + \min\left\{\sigma' + \rho'\left\|[t',t]\right\|, \pi'\left\|[t',t]\right\|\right\} - \left(\left\|[t',t]\right\| - 1\right) \\ &= 3L_{\max} + \min\left\{\sigma' + (\rho'-1)h, (\pi'-1)h\right\} + 1 \end{aligned}$$

where $h = \left\|[t',t]\right\|$. Note that the function $\min\left\{\sigma' + (\rho'-1)h, (\pi'-1)h\right\}$ is the minimum of two linear functions of *h*, where one is nonincreasing and the other is nondecreasing. The function is maximized when $\sigma' + (\rho'-1)h = (\pi'-1)h$, which occurs when $h = \frac{\sigma'}{\pi' - \rho'} = \frac{\sigma}{\pi - \rho}$.

Then

$$\begin{aligned} g(t) &\leq 3L_{\max} + (\pi'-1)\frac{\sigma}{\pi-\rho} + 1 \\ &= 3L_{\max} + \left(\frac{\pi}{W} - 1\right)\frac{\sigma}{\pi-\rho} + 1 \\ &\leq d_{\min} \end{aligned}$$

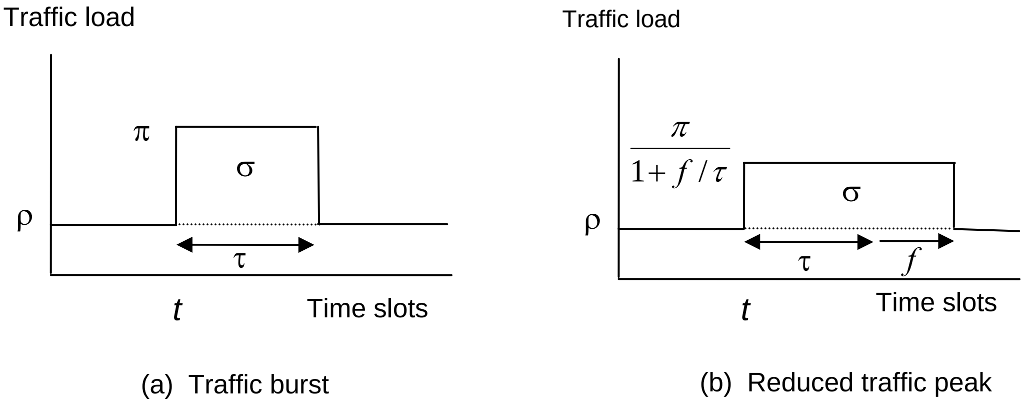
where the last inequality is implied by the Inequality (2.6). Since $g(t) \le d_{\min}$, packet i is scheduled within its spread. Since packet i is arbitrary, all packets are scheduled within their spreads, and the first come first served assignment is a feasible assignment. ■

Corollary 2.1. Suppose there is a batch of lightpath requests that are (σ,ρ,π) constrained. The requests have the same time flexibility *f.* Let the durations of the requests be at least L_{mim} and at most $L_{\max}$. Suppose $f > 3L_{\max} - L_{\min} + 1$. Suppose $\pi \ge W \ge \rho$, and inequality (2.4) is held, and

$$W \ge \frac{\pi}{1+\bar{f}/\tau} \qquad (2.7)$$

where $\bar{f} = f + L_{\min} - 3L_{\max} - 1$ and $\tau = \dfrac{\sigma}{\pi - \rho}$. Then there is a feasible assignment for the batch of lightpath requests.

We have the following interpretation for the formula in (2.7). To simplify the discussion we will assume that the lightpath durations are negligible so that $\bar{f} = f$. We consider lightpath requests which occur as a traffic burst at peak load π starting from some time *t*. The burst can remain at the peak load for a limited time length τ due to the (σ,ρ) parameters. This is illustrated in Figure 2(*a*), which shows the distribution of traffic over time. The burst is a rectangle with width τ. If there is no time flexibility then, to support the traffic distribution, the number of wavelengths *W* must be at least the peak of the traffic distribution, i.e., the height of the rectangle. If there is time flexibility *f* then the traffic can shift to the right by *f*, as shown in Figure 2(*b*). This reduces the traffic peak, and the number of wavelengths *W*, to $\dfrac{\pi}{1+f/\tau}$.



(a) Traffic burst

(b) Reduced traffic peak

Figure 2. Traffic distribution sensitivity with *f*.

The formula $\frac{\pi}{1+f/\tau}$ suggests that to significantly reduce wavelengths, the time flexibility *f* must be at least a significant fraction of τ, the maximum width of a burst at peak load. Note that the formula is an upper bound for all traffic satisfying the (σ,ρ,π) constraints and is therefore conservative. If the actual traffic has smaller bursts then the number of wavelengths may be more sensitive to *f*. In addition, the bursts may have a different distribution than the rectangle shown in Figure 2. For example, if the bursts have slopes to their sides (e.g., trapezoidal or triangular bursts) then the top of the bursts have a smaller traffic-mass than their base. Since the top has a smaller traffic-mass, it takes a smaller *f* (to widen the base) to reduce the height of the burst. For these traffic distributions, the number of wavelengths is more sensitive to *f*.

3 Wavelength Efficiency: Multiple Time Flexibilities

We consider multiple time flexibilities. To model this, we assume that the traffic is composed of *M* traffic types, for some integer *M*, and each traffic type *k* has the same time flexibility f_k. In addition, each traffic type *k* is $(\sigma_k, \rho_k, \pi_k)$ constrained, where σ_k, ρ_k, and π_k are some parameter values.

A simple method to determine the number of wavelengths for this traffic is to apply Corollary 2.1 to each traffic type. Then for each traffic type *k*, we will have a number of wavelengths W_k to support only traffic type *k*. If we use the right hand side of (2.7) for W_k then $W_k = \frac{\pi_k}{1+\bar{f}_k/\tau_k}$, where $\bar{f}_k = f_k + L_{\min} - 3L_{\max} - 1$ and $\tau_k = \frac{\sigma_k}{\pi_k - \rho_k}$. Then the total number of wavelengths required for the overall traffic is $\sum_{k=1}^{M} W_k$.

We can further simplify the traffic model to lead to a rule of thumb for pricing lightpath services. The simplification is to assume that the $(\sigma_k, \rho_k, \pi_k)$ parameters have the same proportion over different traffic types. In particular, there are values σ, ρ and π such that for each traffic type *k*, there is a constant α_k where $\sigma_k = \alpha_k \sigma$, $\rho_k = \alpha_k \rho$, and $\pi_k = \alpha_k \pi$. Then $W_k = \alpha_k \frac{\pi}{1+\bar{f}_k/\tau}$, where $\tau = \frac{\sigma}{\pi - \rho}$. If α_k is proportional to the number of lightpath requests of type *k* then we would expect that the price for lightpath service be proportional to $\frac{1}{1+\bar{f}_k/\tau}$.

However, this estimate for the number of wavelengths can be inaccurate and conservative because there was no resource sharing across different traffic types. For the rest of this section, we will present a more accurate but more complicated estimate. To simplify the discussion, we will make the following assumptions. First, for each traffic type *k*, π_k is assumed to be essentially infinite. Thus, we ignore a peak rate and the traffic is just

(σ_k, ρ_k) constrained. Second, *T* is very large and all lightpath durations are negligible compared to time flexibilities. Then the traffic can be approximated by a *fluid model*. Also note that the spread of a lightpath request is equal to its time flexibility. We also assume that *W* is large enough to accommodate all traffic, i.e., $\frac{1}{T}\sum_{n=0}^{T-1} A(n) < W$.

Since the traffic can be approximated by a fluid model, it can be distributed evenly among all wavelengths. Then the virtual packet-switched queueing system with *W* transmission links, each with a unit of bandwidth, can be approximated by a system with one transmission link with *W* units of bandwidth.

Lemma 2.2 applies to this queueing system with one transmission link. Then, for any work conserving service policy, there is a starting time slot *s* that does not lead to work overlap. Therefore, we can treat the virtual packet-switched, queueing system as an ordinary packet switched system over the time interval [*s*, *s*-1]. Let us assume that the work conserving scheduling policy is *earliest deadline first*. Then we can apply the results in [18], which provides upper bounds on the delays in the virtual packet-switched, queueing system. Under our assumptions, these upper bounds correspond to time flexibilities. Then we have the following conditions for each traffic type *m*,

$$f_m \geq \frac{\sigma_m + \sum_{i=1}^{m-1}(\sigma_i - \rho_i f_i)}{W - \sum_{i=1}^{m-1}\rho_i}$$

This can be rewritten as

$$W \geq \frac{\sum_{i=1}^{m}\sigma_i + \sum_{i=1}^{m-1}\rho_i (f_m - f_i)}{f_m} . \tag{3.1}$$

If these conditions are satisfied for each traffic type *m* then there is a feasible assignment for the traffic.

We now discuss the implications of (3.1). To simplify the discussion further we consider only two traffic types (i.e., $M = 2$). Then

$$W = \max\left\{\frac{\sigma_1}{f_1}, \frac{\sigma_1 + \sigma_2}{f_2} + \rho_1\left(1 - \frac{f_1}{f_2}\right)\right\} \tag{3.2}$$

For comparison, the formula in (2.7) reduces to σ / f under the simplifying assumptions.

There are two cases to discuss.

Case 1, $W = \frac{\sigma_1}{f_1}$*, i.e., traffic of type 1 dominates*:

Then $W = \alpha_1 \dfrac{\sigma}{f_1}$. If $\alpha_1$is proportional to the number of lightpath requests of type 1 then each request contributes to *W* by an amount that is proportional to$1/f_1$. Then it is reasonable for the requests to have prices proportional to $1/f_1$. Customer prices for lightpath requests of type 2 can be small since they have no affect on network resources. We can also change f_2 without affecting *W.*

This case is illustrated in Figure 3. Figure 3(*a*) is the traffic distribution when each traffic type k has a burst σ_k at time *t,* and average load ρ_k thereafter. Traffic type *k* can be shifted to the right by f_k which will reduce the peak traffic. The shifted traffic distribution is shown in Figure 3(*b*), and recall that the number of wavelengths is dependent on the peak traffic. Note that the burst σ_1 for traffic 1 has a distribution of a rectangle with a width of f_1, and it contributes to the peak traffic of $\dfrac{\sigma_1}{f_1}$. Since the burst σ_2 for traffic 2 is small, it can be shifted beyond the burst of type 1 without contributing to the peak traffic.

Case 2, $W = \dfrac{\sigma_1 + \sigma_2}{f_2} + \rho_1\left(1 - \dfrac{f_1}{f_2}\right)$ *i.e., traffic of type 1 does not dominate*:

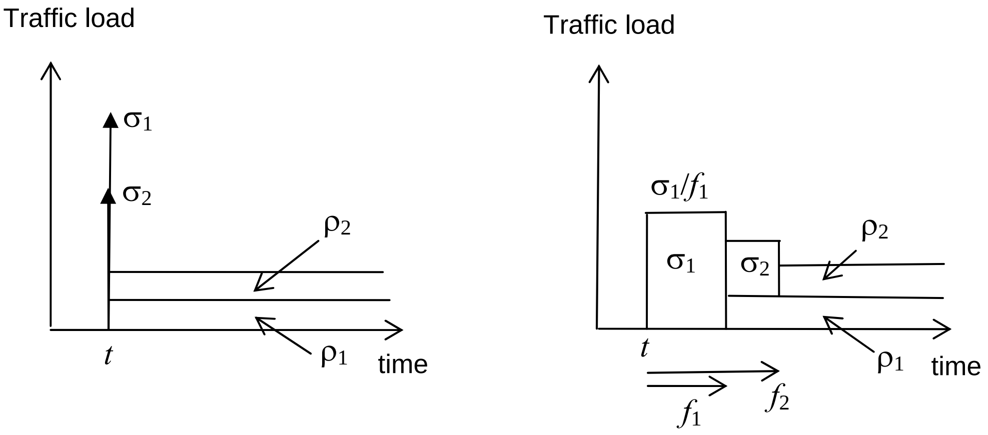
It is straightforward to rewrite *W* as

$$= \frac{\sigma_2}{f_2} + \sigma_1\left(\frac{1}{f_2} + \left(\frac{\rho_1}{\sigma_1}\right)\left(1 - \frac{f_1}{f_2}\right)\right)$$

$$= \alpha_2\frac{\sigma}{f_2} + \alpha_1\sigma\left(\frac{1}{f_2} + \left(\frac{\rho}{\sigma}\right)\left(1 - \frac{f_1}{f_2}\right)\right)$$

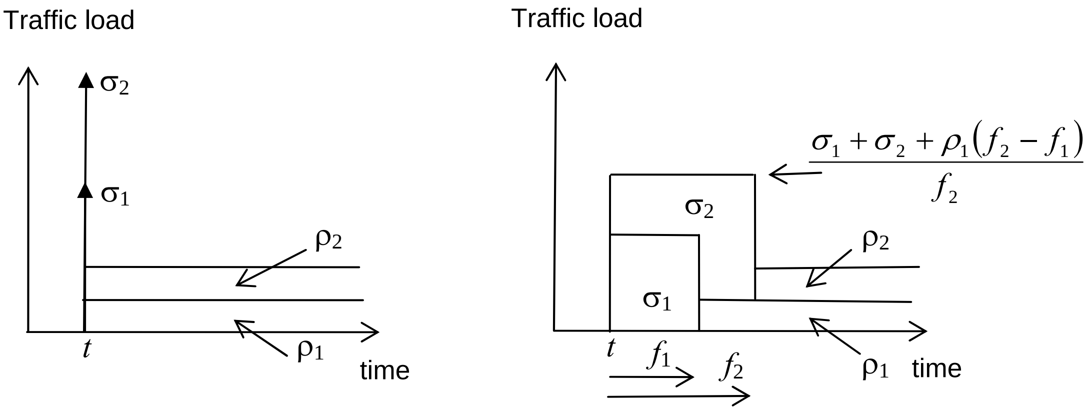
Suppose, respectively, α_1 and α_2 are proportional to the number of lightpath requests for, respectively, traffic of type 1 and 2. Then customer prices for lightpath requests of type 2 should be proportional to $1/f_2$. Customer prices for lightpath requests of type 1 should be proportional to $\dfrac{1}{f_2} + \left(\dfrac{\rho}{\sigma}\right)\left(1 - \dfrac{f_1}{f_2}\right)$. Notice that the prices are higher than for type 1 but the difference becomes smaller as the traffic becomes burstier or the flexibilities become the same.

This case is illustrated in Figure 4. Figure 4(*a*) is the traffic distribution when each traffic type k has a burst σ_k at time *t*, and average load ρ_k thereafter. Traffic type *k* can be shifted to the right by f_k which will reduce the peak traffic. The shifted traffic distribution is shown in Figure 3(*b*), and recall that the number of wavelengths is dependent on the peak traffic. The burst σ_1 for traffic 1 has a distribution of a rectangle with a width of f_1. The burst σ_2 for traffic 2 is large, so to minimize the peak traffic, it must be distributed as shown in Figure 4(*b*). In particular, it must be distributed so that the overall traffic load over the time interval $[t, t + f_2]$ is flat. During this interval, there is the burst σ_1, the burst σ_2, and average traffic load ρ_1, that have been shifted from time t to time $t + f_1$. The total traffic in this interval is $\sigma_1 + \sigma_2 + \rho_1(f_2 - f_1)$. This total traffic divided by the interval length f_2 is the peak traffic.



(*a*) Traffic distribution. (*b*) Distribution after shifting.

Figure 3. Traffic distribution dominated by traffic of type 1.



(*a*) Traffic distribution. (*b*) Distribution after shifting.

Figure 4. Traffic distribution when traffic is not dominated by traffic of type 1.

4 Algorithms and Performance

We ran simulations to study how wavelengths depend on time flexibility for randomly generated traffic. The problem of determining the number of wavelengths given a batch of lightpath requests is an NP Complete problem. In particular, a special case of the problem is when the requests have no restrictions on time flexibility, i.e., the time flexibility is T–1. Then the problem is the bin packing problem [19], where the wavelengths are the bins, T is the size of each bin, and the requests are the objects to be packed in the bins. The bin packing problem is NP Complete. Therefore, since we are considering large instances, we use heuristics to determine the number of wavelengths. The heuristics are described in Subsection 4.1. Our simulations and performance results are presented in Subsection 4.2.

4.1 Heuristics

The following are the heuristics we considered. The first set of heuristics use the virtual packet-switched queueing system model as described in Section 2.1, where the lightpath requests are virtual packets and wavelengths correspond to the transmission links. For a given number of wavelengths W, the basic computation of the heuristic is as follows. Virtual packets are scheduled starting from some time slot s in the time interval [s, s–1], if $s > 0$, or [0, T-1], if $s = 0$. To simplify our discussion, we will assume without loss of generality that $s > 0$, and so the interval is [s,s–1]. The scheduling algorithms we considered were *first come first served* (*FCFS*) and *earliest deadline first* (*EDF*). In the case of FCFS, a virtual packet (a, b, L) with the smallest “arrival time” a is scheduled first. In the case of EDF, a virtual packet (a, b, L) with the smallest “deadline” b+L is scheduled first. In both cases, a scheduled virtual packet is assigned to the lowest available wavelength within its allowed spread.

As the scheduling reaches the end of the time interval [s, s–1], there may be virtual packets that are queued that require time slots that wrap around to the beginning of the time interval. They are assigned to wavelengths with the available time slots. Note that these virtual packets create a burst of traffic load around time s which we refer to as the *wrap around traffic load*. This can limit wavelength efficiency.

Our algorithms consider each value of s until it finds a feasible assignment. There may be no feasible assignment if W is too small. The aforementioned heuristics assign lightpath requests by following the time interval [s, s–1], and assign requests to the available wavelength. The next three heuristics use a different assignment approach by filling one wavelength at a time, starting from wavelength 0, and assign request to available time slots.

For the *Lowest Wavelength, Maximum Duration* (*LWMD*) heuristic, to fill wavelength k, lightpath requests that have longer durations are considered first, where ties are broken randomly. Preference is given to requests with longer durations because they are more difficult to assign. To fit a lightpath request (a,b,L) into the wavelength, start times are considered in the start interval $[a,b]$ beginning with a.

For the *Lowest Wavelength, Fixed* (*LWFixed*) heuristic, an initial time *s* is chosen. It fills one wavelength at a time, starting from wavelength 0. It fills each wavelength *k* by starting from the same initial time *s* as follows. Without loss of generality, we assume *s* > 0. Choose the longest unassigned request (a, b, L) that could start at time *s* and assign it starting from *s*. Continue filling the wavelength from time $s + L$ (here, the addition is modulo *T*). If there is no such request then continue filling the wavelength from time *s*+1. This continues until we reach the end of the interval [*s*, *s*−1]. When all requests are assigned, then the number of wavelengths *W* is noted. Note that since LWFixed always fills wavelengths from the same initial time, it creates a wrap around traffic load like EDF and FCFS. The next heuristic tends to avoid this.

The *Lowest Wavelength, Continuous* (*LWCont*) heuristic is similar to LWFixed. It fills wavelength 0 in the same way by starting from time 0. However, it fills wavelength *k* > 0 by starting at a time *t* that depends on how wavelength $k-1$ was filled. In particular, if wavelength $k-1$'s last request was assigned time slots $[x, y]$ then wavelength *k* is filled starting from $y+1$ (here, the addition is modulo *T*). In this way, the assignment for wavelength *k* continues from where the assignment for wavelength $k-1$ ended.

The last heuristic is the assignment algorithm described in Section 2.2. The traffic is first split among the wavelengths using the load balancing algorithm of Subsection 2.2. For each wavelength, its traffic is assigned time slots using the algorithm described in Subsection 2.2 where the work conserving policy is *earliest deadline first*. We will refer to this as the *Load Balance Then Schedule* (*LBTS*) algorithm.

4.2. Simulations

We simulated the algorithms in Subsection 4.1 for the following traffic. Each time slot is a 10 minute period. The number of time slots is *T* = 144, which is the number of slots in a 24 hour period. There is a fixed number of lightpath requests *R* with the same flexibility *f*. The durations of the lightpath requests were chosen either (*i*) randomly, independently, and uniformly distributed on the interval [1, *δ*-1], where *δ* is a parameter; or (*ii*) all have the same fixed duration . Then the average duration L_{avg} = *δ*/2 if the durations are random, or L_{avg} = *δ* if the durations are fixed.

Let {s_1, s_2, ..., s_R} denote the start times of the lightpath requests. These start times are independent and identically distributed. We consider three distributions. The first is the uniform distribution on the interval [0, *T*-1], i.e., it has a density function $p(t) = \frac{1}{T}$ for all $t \in [0, T-1]$. The other two distributions model a single peak of traffic. The second distribution has a single uniform burst on the interval [*T*/3, 2*T*/3]. We refer to it as the *rectangular* distribution. It has the following density function

$$p(t) = \begin{cases} 2.2/(1.4\,\mathrm{T}) & \text{if} \quad T/3 \le t < 2T/3 \\ 1.0/(1.4\mathrm{T}) & \text{otherwise} \end{cases}$$

The third distribution is a modified Gaussian distribution with mean $\mu = T/2$ and variance σ. We describe its density function p. For each time slot t in [0, T-1], there is a function to model the distribution of request start time $g(t) = \exp\left(-(t+0.5-\mu)^2 / 2\sigma^2\right)$. (Note that t+0.5 is the time instant in the middle of time slot t.) The start times of the lightpath requests are distributed according to the following density function

$$p(t) = \begin{cases} g(t)/G & \text{if} \quad 0 \le t < T \\ 0 & \text{otherwise} \end{cases}$$

where $G = \sum_{t=0}^{T-1} g(t)$ is a normalization constant.

The simulations measured the number of wavelengths required for each heuristic algorithm as shown in Figures 5, 6, 8 and 10. For EDF, FCFS, and LBTS, the algorithms were applied to different numbers of wavelengths. The wavelength chosen was the smallest that allowed all lightpath requests to be scheduled. The figures show the average number of wavelengths over 10 randomly generated batches of traffic.

In addition, the figures present three additional curves. The first curve is the constant $\frac{RL_{avg}}{T}$, which is the statistical mean of the traffic load averaged over the time slots. We refer to this as the "Statistical Lower Bound" because it is a lower bound on the number of wavelengths if the total traffic load equals its statistical mean. The second curve is the upper bound of Corollary 2.1, which is computed as follows. For each random batch of simulation traffic, the values (σ,ρ,π) are computed as $\rho = \frac{1}{T}\sum_{t=0}^{T-1} A(t)$, $\pi = \max_t A(t)$, and σ is the smallest value that satisfies $\sum_{t \in [x,y]} A(t) \le \min\{\sigma + \rho|[x,y]|, \pi|[x,y]|\}$ over all time intervals [x,y]. Then

$$W = \left\lceil \max\left\{ \frac{\pi}{1 + \bar{f}/\tau}, \rho\left(1 + \frac{3L_{\max}}{T - 3L_{\max}}\right) \right\} \right\rceil \tag{4.1}$$

is calculated as described in the corollary. The figures plot the average of W over the 10 batches of traffic, and only when $f > 3L_{\max} - L_{\min} + 1$. We refer to this as the "Empirical Upper Bound".

The third curve is also computed using the formula (4.1), but it assumes that the traffic load at each time t is exactly equal to the statistical mean at that time, i.e., $A(t) = p(t)L_{avg}R$. We refer to this as the "Statistical Upper Bound". Note that this may not be an upper bound for every random batch of requests. Also note that computing this statistical upper bound curve only depends on the expected traffic load, which is a prediction of the actual traffic load. We shall see if the bound is an accurate predictor of the number of wavelengths.

Next we present results of simulations performed on traffic requests with random durations. Figure 5 shows the number of wavelengths when the starting times are uniformly distributed, δ = 24, and R = 288. The average number of wavelengths is plotted versus time flexibility. The LWCont, LWFixed, and LWMD require the least wavelengths, where LWCont and LWFixed are almost identical. Notice that modest amounts of flexibility result in significant reduction in wavelengths. When the flexibility is 24, the average number of wavelengths is within two of the lower bound of 24. The other algorithms EDF, FCFS, and LBTS are inefficient. EDF and FCFS are inefficient because of the wrap around traffic load. LBTS is inefficient because it uses a simple load balancing rule, which will likely create a high load on a wavelength. We do not show the empirical or statistical upper bounds because they are quite loose with an average value of 48. The statistical lower bound is the value 24.

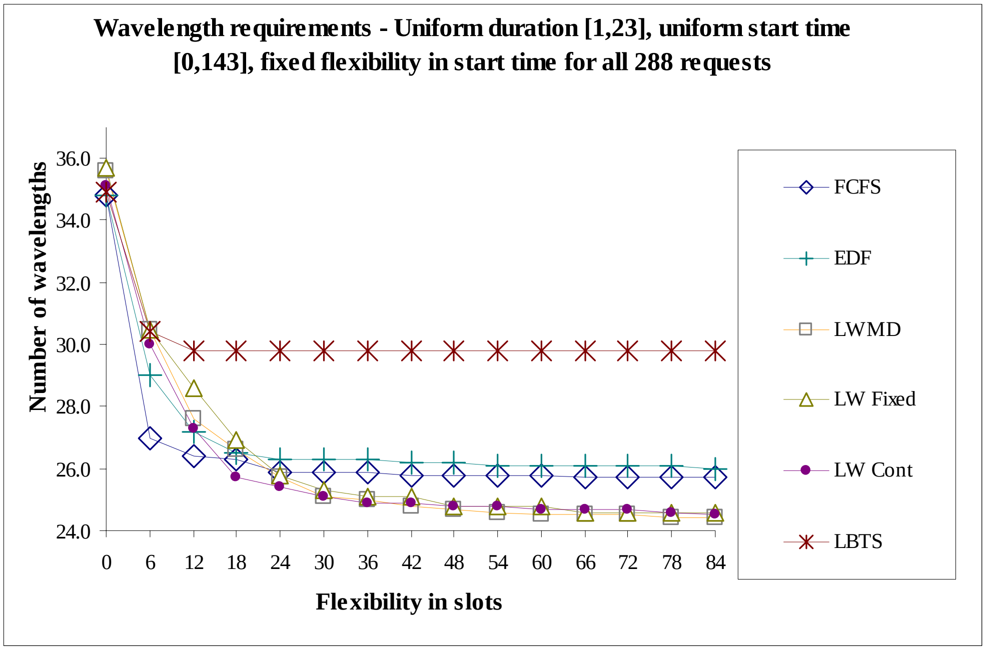


Figure 5. Uniform distribution in start times, δ = T/6 = 24 and R = 288.

Figure 6 presents the results for the smaller value of δ = 3 and R = 2016. Then $L_{\min} = 1$ and $L_{\max} = 2$. Now the algorithms have similar performance, and the upper bounds are better estimates of performance.

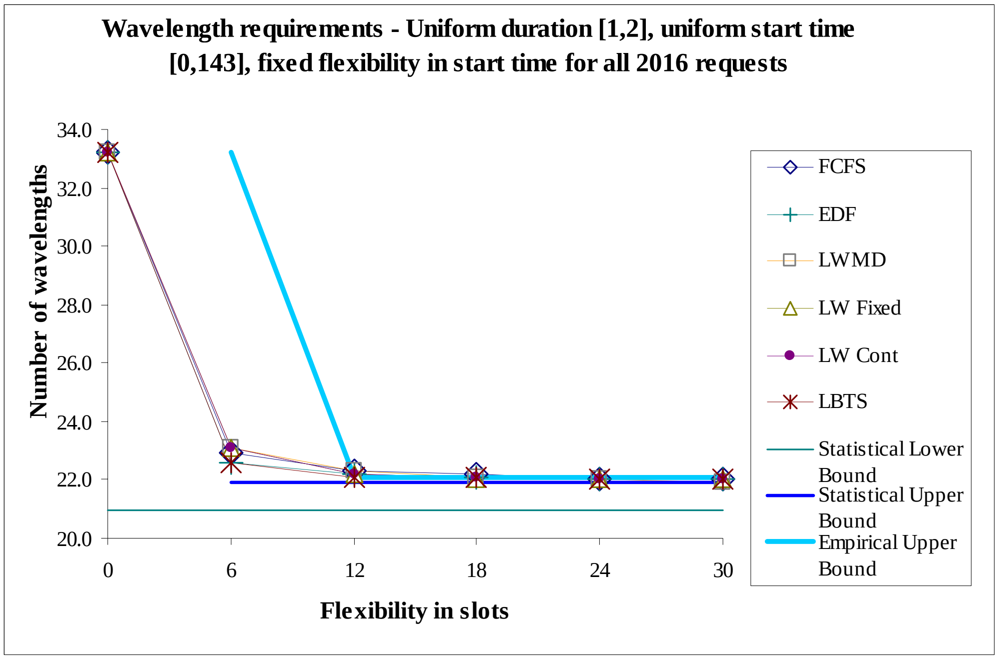


Figure 6. Uniform distribution in start times, δ = *3* and *R* = 2016.

Next we consider simulations for a rectangular distribution of start times when $\delta = 3$ and R = 2016. The distribution has a single uniform burst on the interval [T/3, 2T/3] (e.g., peak hours 8 AM to 4 PM) at an average traffic load of 33. Outside the interval, the average traffic load is 15. Figure 7 presents the traffic load over time for a sample batch of traffic and the statistical mean of the traffic load, i.e., $p(t)L_{avg}R$. Note that the sample batch is much burstier than the statistical mean.

Figure 8 presents results for the rectangular distribution of traffic. FCFS, EDF, and LBTS have similar performance, and perform better than LWMD, LWFixed, or LWCont. In the case of FCFS and EDF, the shape of the rectangular distribution, with a large peak and a large valley eliminates the effect of wrap around traffic load. In particular, if the wrap around traffic load is in the valley, it will have no effect on the number of wavelengths, which is determined by the peak. In the case of LBTS, since the lightpath durations are small, the traffic is split evenly among the different wavelengths and leads to efficient wavelength use. Note that LWMD, LWFixed, and LWCont have almost identical performance.

For this traffic, the statistical upper bound is an accurate estimate of the number of wavelengths. This may be explained by the rectangular shape of the distribution of the statistical mean. Since the shape is similar to Figure 2, it leads to the performance of Corollary 2.1 and formula (4.1). The empirical upper bound is more conservative because its (σ, ρ, π) values were computed for a burstier traffic.

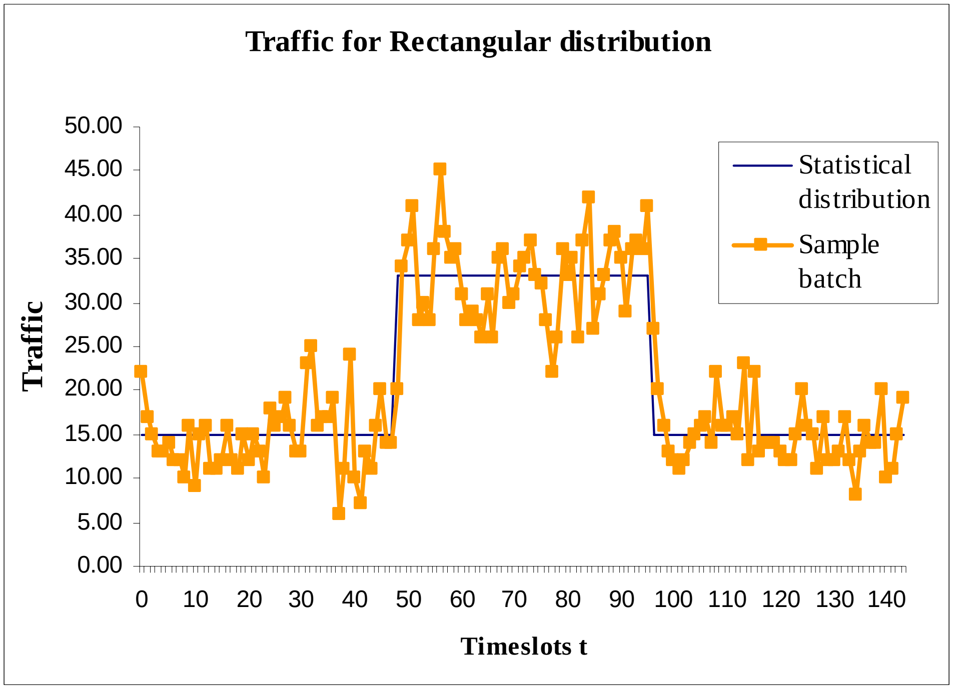


Figure 7. Traffic for Rectangular distribution.

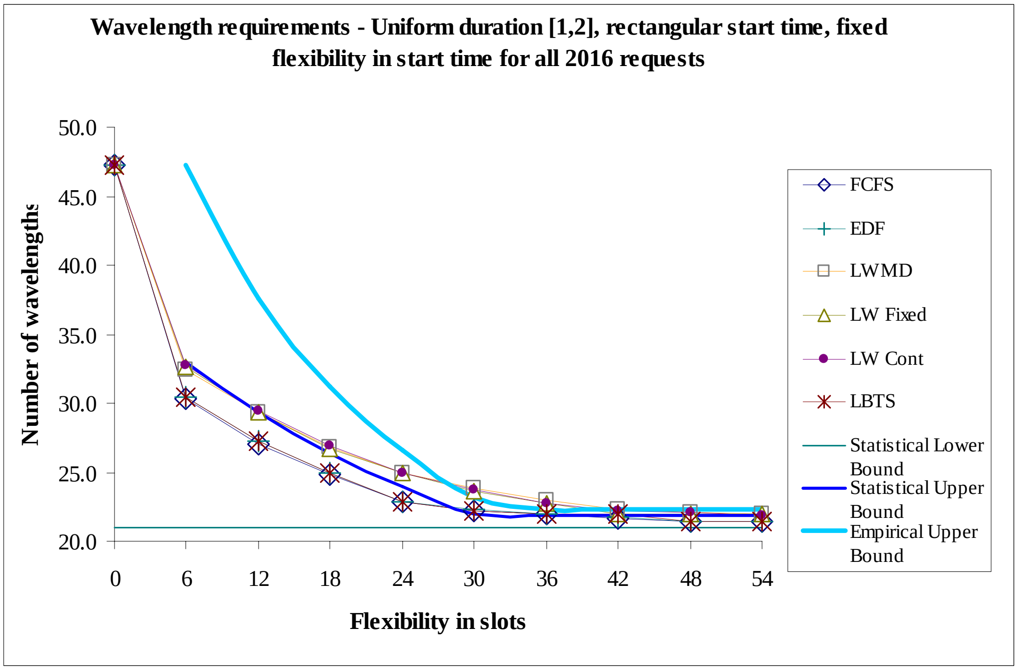


Figure 8. Rectangular distribution in start times, δ = *3* and *R* = 2016.

Next we consider simulations for the modified Gaussian distribution for start times when $\delta = 3$ and R = 2016. Note that there is a peak traffic period during the day, e.g., during working hours. The statistical distribution has a mean of *T*/2, which is 72 slots of 10 minutes each or 12 hours (noon). The standard deviation is set at *T*/6, or 4 hours. Figure

9 presents the traffic load over time for a sample batch of traffic and the statistical mean of the traffic load.

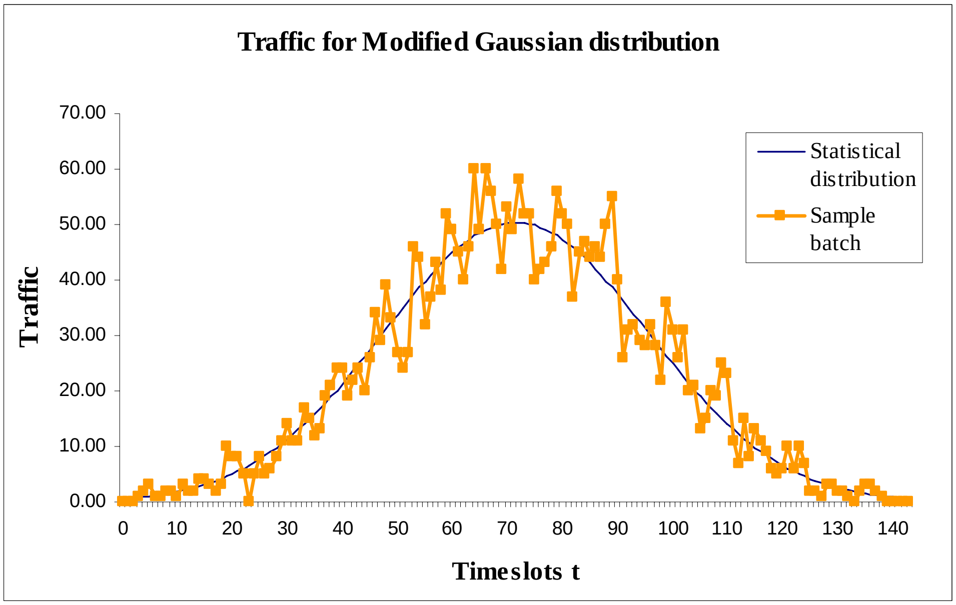


Figure 9. Traffic for Modified Gaussian distribution.

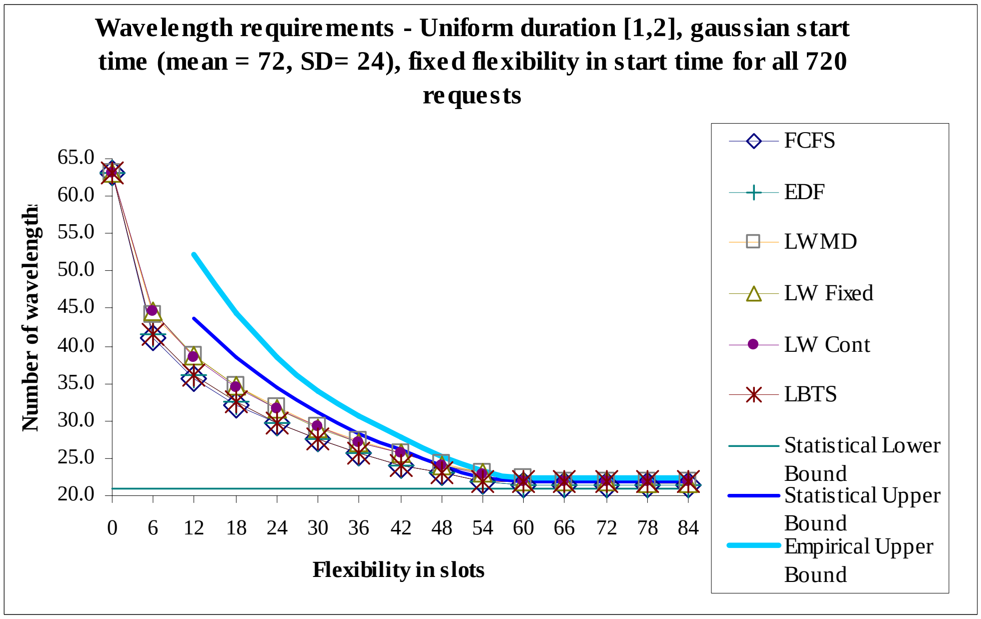


Figure 10. Modified Gaussian distribution in start times, δ = *3* and *R* = 2016.

Figure 10 presents the results of the simulations. The performance of the heuristics is similar to that of the rectangular distribution. This is due to the fact that both the distributions are characterized by a single peak of traffic and a large valley. The statistical upper bound is still a reasonable estimate of the number of wavelengths, even though the burst of the modified Gaussian distribution is not shaped like a rectangle.

Figures 8, and 10 show that the statistical upper bound is a reasonable estimate of performance for traffic with a single peak. But it is for traffic with small request durations, within 20 minutes. In general, when the durations become large, the bound is a poor estimate. Figures 11 and 12 show that the statistical upper bound can still be a reasonable estimate for $L_{avg} = 6$, which corresponds to an hour. But the durations are fixed, which improves the bound. Figures 11 and 12 are for start times that have, respectively, the rectangular and modified Gaussian distributions. In all cases, R = 720. Note that FCFS, EDF, and LBTS perform best. Note that since the lightpath durations are fixed, FCFS and EDF are identical.

The larger durations lead to differentiation of performances of LWFixed and LWCont from LWMD, where LWFixed and LWCont perform better. Note that since all the lightpaths have the same duration, the LWMD becomes less effective.

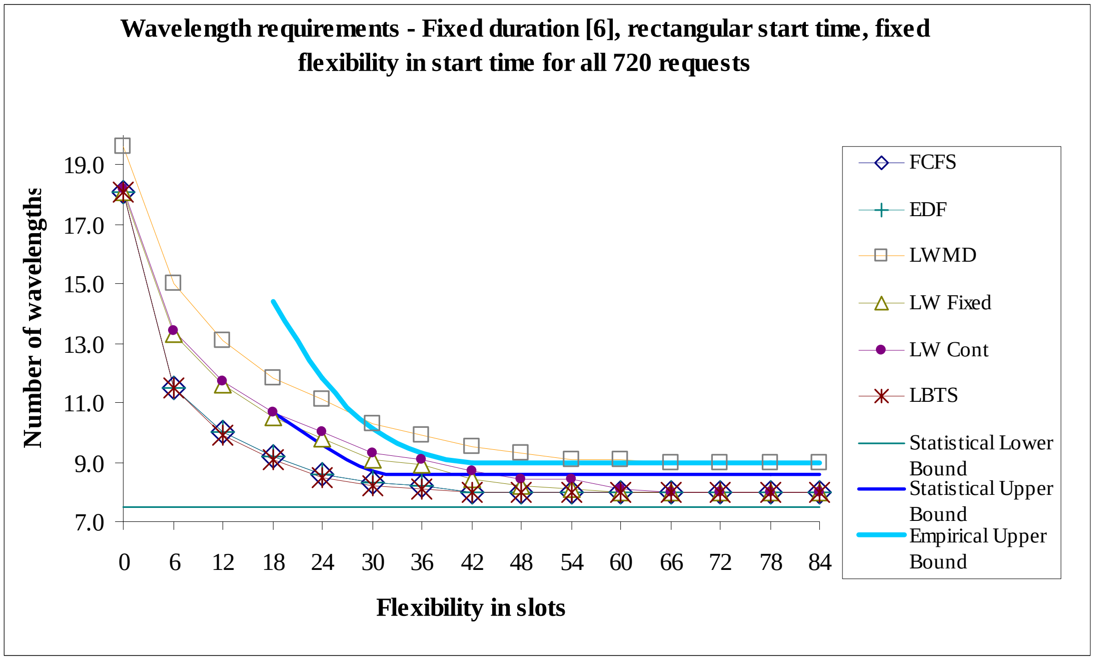


Figure 11. Rectangular distribution in start times, δ = 6 and R = 720.

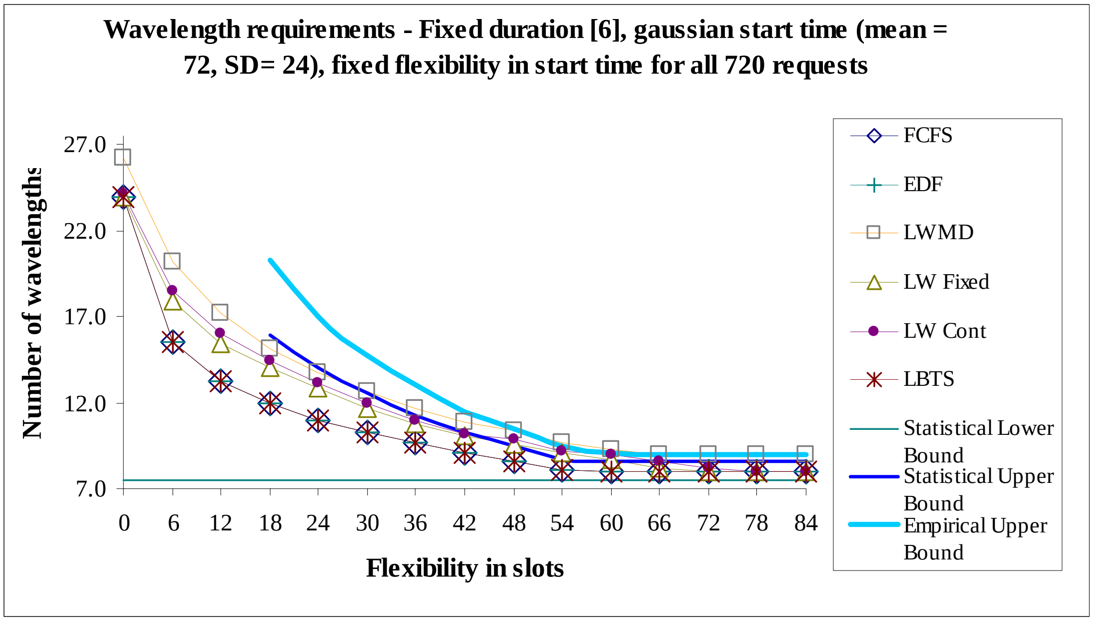


Figure 12. Modified Gaussian distribution in start times, δ = 6 and *R* = 720.

5 Conclusions

We considered scheduled lightpaths to provide periodic optical connectivity. We discussed how time flexibility can reduce the number of wavelengths. By applying the leaky bucket traffic model [1], we were able to derive formulas that show the relationship between wavelengths and traffic parameters and time flexibility.

If the durations of the scheduled lightpaths are relatively short with respect to time flexibility, then the number of wavelengths is approximately $\frac{\pi}{1+f/\tau}$, as described in Section 2. We verified by simulation that this formula is a reasonable estimate for traffic that have a rectangular or Gaussian distribution, which are distributions with a single peak. This is a relatively simple formula which may be useful for pricing. However, further study is required to completely understand how time flexibility affects wavelengths. We demonstrated that traffic with multiple time flexibilities can complicate an accurate analysis. Other directions for future work are to provide more accurate formulas when lightpath durations are large, and extend the analysis to multi-hop networks.

We proposed and simulated a number of practical heuristic scheduling algorithms for a single WDM link. Heuristics were used because the scheduling problem is NP-Complete even for the single link. Our experiments showed that even moderate amounts of time flexibility can dramatically reduce the number of wavelengths. FCFS, EDF, and LBTS work well when lightpath durations are short and for traffic with a single peak. When lightpath durations are long, on average with a mix of lengths, then heuristics that fill low valued wavelengths first work better.

6 Acknowledgements

The authors are grateful for the insightful comments of Dr. Gustavo de Veciana and reviewers.

Reference

[1] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, R. Douville, Routing and wavelength assignment of scheduled lightpath demands, IEEE J. Sel. Areas Commun. 21 (8) (2003) 1231-1240.

[2] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, Routing foreseeable lightpath demands using a tabu search meta-heuristic, in: Proc. IEEE GLOBECOM 2002, Taipei, Taiwan, Nov. 2002, pp. 2803-2807.

[3] W. Su, G. Sasaki, Scheduling of periodic transfers with flexibility, in: Proc. 41st Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, Oct. 1-3, 2003.

[4] H. Zang, J. Jue, B. Mukherjee, A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks, Optical Networks Magazine 1 (1) (2000) 47-60.

[5] E. W. Fulp, D. S. Reeves, The economic impact of network pricing intervals, in: Proc. Adv. Internet Charging and QoS Technology ICQT 2002, Zurich, Switzerland, Oct. 16-18, 2002.

[6] E. W. Fulp, D. S. Reeves, Bandwidth provisioning and pricing for networks with multiple classes of service, Computer Networks: Int. J. of Computer & Telecomm. networking 46 (1) (2004) 41-52.

[7] I. Ch. Paschalidis, J. N. Tsitsiklis, Congestion-Dependent Pricing of Network Services, IEEE/ACM Trans. Networking 8 (2) (2000) 171-184.

[8] R. Cruz, A calculus for network delay, part I: network elements in isolation, IEEE Trans. Info. Theory 37 (1) (1991) 114-131.

[9] S. Subramaniam, E. Harder, H. Choi, Scheduling multirate sessions in time division multiplexed wavelength-routed networks, IEEE J. Sel. Areas Commun. 18 (10) (2000) 2105-2110.

[10] B. Wen, K. Sivalingam, Routing, wavelength and time-slot assignment in time division multiplexed wavelength-routed optical WDM networks, in: Proc. IEEE INFOCOM, June 2002.

[11] A. Chen, A. K.-S. Wong, C.-T. Lea, Routing and time-slot assignment in optical TDM networks, IEEE J. Sel. Areas Commun. 22 (9) (2004) 1648-1657.

[12] R. Guerin, A. Orda, “Networks with advance reservations: the routing perspective, in: Proc. IEEE INFOCOM, vol. 1, 2000, pp. 118-127.

[13] J. Zheng, H. T. Mouftah, Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks, in: Proc. IEEE Int. Conference on Communications ICC 2002, New York, Apr. 2002, vol. 5, pp. 2722-2726.

[14] A. Tucker, Coloring a family of circular arcs, SIAM J. Applied Math. 29(3) (1975) 493-502.

[15] N.M. Sadeh, M.S. Fox, Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem, Artificial Intelligence, 86 (1996) 1-41.

[16] L.-O. Burchard, On the performance of computer networks with advance reservation mechanisms, in: Proc. 11th IEEE Int. Conference on Networks ICON 2003, Sydney, Australia, 2003.

[17] B. Wang, T. Li, X. Luo, Y. Fan, Traffic grooming under a sliding scheduled traffic model in WDM networks, in: Proc. IEEE Workshop on Traffic Grooming in WDM Networks, San Jose, Oct. 29, 2004.

[18] J. Liebeherr, D.E. Wrege, D. Ferrari, Exact admission control for networks with a bounded delay service, IEEE/ACM Trans. Networking 4 (6) (1996) 885-902.

[19] E. G. Coffman, M. R. Garey, D. S. Johnson, Approximation algorithms for bin packing: a survey, in: D. Hochbaum (Ed.), Approximation algorithms for NP-hard problems, PWS Publishing, Boston, 1996, pp. 46-93.