Scheduling of Periodic Connections with Flexibility[1]

Wei Su and Galen Sasaki
University of Hawaii
2540 Dole Street, Honolulu, HI 96822 USA
{weisu, sasaki}@spectra.eng.hawaii.edu

Abstract: A wavelength division multiplexed link is to be time shared by a given collection of optical connections. The system is time slotted with period *T*. Each optical connection has a prescribed duration per time period. In addition, it has a prescribed time window within which its duration may begin. The size of the window is the amount of time flexibility there is to schedule the optical connection. The problem considered is to set up the optical connections. Formulas are provided to show the relationship between wavelength efficiency and the "burstiness" and time flexibility of the connections. In addition, simple heuristic algorithms are presented that perform well under simulations for randomly generated optical connections.

1. Introduction

Wavelength division multiplexing (WDM) technology allows fiber-optic links to carry multiple wavelength channels, each capable of transporting tens of gigabits per second. A *wavelength routed network* deploys WDM technology to carry *lightpaths,* which are end-to-end optical connections. If the network is *all optical* then lightpaths can carry signals with arbitrary signal formats, e.g., SONET or Gigabit Ethernet. This is known as *optical transparency* and it provides greater flexibility to users. However, many applications may need an optical connection for only short periods of time, and leasing a lightpath full time for that purpose may be too costly.

In this paper, we consider networks where wavelengths are shared by time division multiplexing. They support users that require lightpaths periodically, say once per day. For example, a client may have built an IP network using lightpaths as IP links. The IP network has a baseline topology that provides minimal network capacity at all times. The links are realized by static lightpaths. However, during working hours of 8am to 5pm when there is more traffic, the IP network needs additional links. These additional links can be realized by lightpaths with periodic service.

As another example, an office may require a Gigabit Ethernet connection to a remote data storage facility to transfer back up data. It may need a lightpath service for 30 minutes per day between 12 midnight and 4 am. Thus, there is a 3.5 hour window of flexibility within which the service can begin.

To study periodic lightpath services, we consider a simple scenario. There is a single fiber-link with W wavelengths. Time is slotted with T time slots per day, and over a period of a day, the time slots are numbered 0, 1,..., T–1. 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 $[0, T-1]$. Note that this implies $s \le t$. If $s,t \in [0, T-1]$ and $s > t$ then $[s,t]$ is

[1] Presented at the Forty-First Annual Allerton Conference on Communication, Control, and Computing, October 1-3, 2003, Monticello, Illinois USA

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 *periodic lightpath service* (or simply *lightpath service*) 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 $L \le T$. 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 as the, respectively, *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 $s \in [a,b]$.

The problem we consider is as follows. Given a batch of lightpath requests, assign them lightpath services so that at any time there is at most one lightpath service per wavelength. We will denote the longest duration of the batch by L_{max}.

An example is shown in Figure 1 when *W* = 2 and *T* = 8. It shows an assignment of the lightpath requests (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.

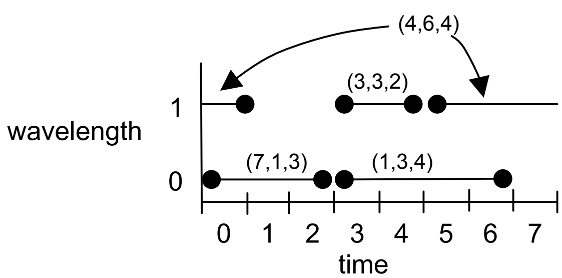

Figure 1. An assignment of a batch of four lightpath requests.

If the requests have no time flexibility then the problem is the same as wavelength assignment of lightpaths on a WDM ring with no wavelength conversion (e.g., see [2] [3] [4]). For the problem, time slots correspond to the nodes of the WDM ring.

In Section 2, we provide simple analytical results to show a relationship between wavelength efficiency, time flexibility, and the "burstiness" of lightpath requests. We use a packet traffic model of Cruz [1] to model "burstiness" of requests. In Section 3, we examine heuristic assignment strategies for randomly generated lightpath requests. We propose several heuristics and compare their performance by simulation. Our final remarks are given in Section 4.

2. Tradeoff Between Wavelengths and Time Flexibility

We will apply a traffic model introduced by Cruz [1] to characterize a batch of lightpath requests. This model leads to formulas that relate time flexibility, burstiness of the requests, and wavelength efficiency. Since this model is for packet traffic, we view the batch of lightpath requests as *virtual packets* as follows. For simplicity, we refer to the sum of durations of a set of lightpath requests as the *work* for the set.

Virtual Packet (VP) Model. 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*. It can be delayed no more than $\|[a,b]\| - 1$ time units, i.e., the time flexibility of the lightpath request. For each time *t*, let $A(t)$ denote the work of the virtual packets that arrive at time *t*. ■

Cruz [1] characterized packet traffic with two parameters (σ, ρ), where ρ is a measure of the average traffic rate, and σ is a measure of the traffic burstiness. We say that the lightpath requests are (σ, ρ) *constrained* if for all time intervals [*x*, *y*],

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

We also assume that $\rho \leq W$ so that the requested bandwidth will tend to be within the capacity of the fiber-link. This does not guarantee that the requested bandwidth will be within the capacity because σ can be larger than 0.

The main results of this section are in two theorems that describe conditions under which an assignment can be found for the batch of lightpath requests. The first assumes that lightpaths do not wrap around at the end of $[0, T-1]$. This makes it easier to find assignments for the lightpath requests. The second theorem is for the general case when lightpaths can wrap around.

For the results, we will refer to the VP Model and the following *First Come First Served* (FCFS) assignment of lightpath requests. FCFS considers time slots in sequence starting from 0. At time *t*, both arriving and "queued" virtual packets (i.e., virtual packets that arrived before time *t* but are still unassigned) are assigned to available wavelengths with preference to those that arrived earliest. When a virtual packet (a, b, L) is assigned to a wavelength *w*, its lightpath service starts at time *t* and has duration *L*. Unassigned virtual packets are queued for time slot *t*+1.

The theorems are stated next and then their proofs.

Theorem 1. Suppose the batch of lightpath requests are (σ, ρ) constrained. Let

$$f = L_{\max} + \left\lfloor \frac{\sigma - L_{\max} + 1}{W} \right\rfloor .$$

Suppose $A(t)=0$ for all $t \in [T-(f+L_{\max}), T-1]$, i.e., there are no virtual packet arrivals for the last $f+L_{\max}-1$ time slots in $[0,T-1]$. (This last condition insures that lightpaths will not wrap around at the end of $[0,T-1]$.) Then there is an assignment for the lightpath requests if their time flexibilities are at least f.

Theorem 2. Suppose the lightpath requests are (σ,ρ) constrained, and $\sum_{t=0}^{T-1} A(t) \le W(T-L_{\max})$. Let $f = \left\lfloor 3L_{\max} + \frac{\sigma}{W} + \frac{\rho}{W} - 1 \right\rfloor$. Then there is an assignment for the batch of lightpath requests if their time flexibilities are at least f.

Before presenting the proofs, we share the following observations. Note that allowing lightpaths to wrap around at the end of $[0,T-1]$ will make it more difficult to find an optimal assignment for lightpath requests. As a result, the required time flexibility f is higher in Theorem 2. Also note that the required time flexibility f is dependent on the burstiness σ of the requests and the number of wavelengths W. Higher burstiness of requests result in higher required time flexibilities and or wavelengths.

Finally, note that the necessity of the constraint $\sum_{t=0}^{T-1} A(t) \le W(T-L_{\max})$ in Theorem 2 can be partly justified through the following example. Let k and L be integers. Suppose $T = k \cdot L - 1$ and all lightpath requests have duration L. Then at most $k-1$ lightpath services can be assigned to a wavelength. The $k-1$ lightpath services of duration L correspond to work equal to $(k-1)L = T-(L-1)$. Thus, the total amount of work that can be assigned to the wavelengths is at most $W(T-(L-1))$.

Proof of Theorem 1. We use the VP Model. We assume that all the requests are assigned using FCFS and none of the lightpaths wrap around. At the end of the proof, we will verify that all the lightpaths begin in the prescribed start windows and none of the lightpaths wrap around.

We will consider an arbitrary virtual packet (a,b,L). Let t denote the time when it gets a wavelength. We will show that $t \le a+f$.

In the subsequent discussion, we refer to the system as being *full* if all wavelengths are being used. Let s be the largest time such that $s \le a$ and the system is not full. If $s = a$ then at time a, the system is not full. Then the virtual packet (a,b,L) must have been assigned a wavelength at time a (i.e., $t = a$), and trivially $t \le a+f$.

Now suppose $s < a$. At time s, since the system is not full, there are at most $W-1$ virtual packets, which we refer to as the *residual virtual packets*. Let R denote the amount of work left by the residual virtual packets at time s+1, and note that R is at most $(W-1)(L_{\max}-1)$. The amount of work C accumulated during the time interval [s+1,a] is $R + \sum_{n \in [s+1,a]} A(n)$ which includes the virtual packet (a,b,L).

Next note that, from the definition of s, the system is always full during the interval [s+1,a]. The system is also full during $[a,t-1]$ because the virtual packet (a,b,L) cannot get service before time t. Thus, the system is full during the interval $[s+1,t-1]$. In addition, during

$[s+1,t-1]$, the system it is filled with work *C*. Also, at time *t*, the system must have some work from *C*, and in particular from the virtual packet (a,b,L). Summarizing, during the interval $[s+1,t-1]$, the system is always full of work from *C*, and at time *t* the system has some work from *C*. This implies that work *C* must be greater than $W\|[s+1,t-1]\|$, i.e.,

$$W(t-s-1) < R + \sum_{n\in[s+1,a]} A(n) .$$

Since the lightpath requests are (σ, ρ) constrained and $R \le (W-1)(L_{\max}-1)$, we have

$$\begin{aligned} W(t-s-1) &< (W-1)(L_{\max}-1)+\sigma+\rho(a-s) \\ W(t-a-1+a-s) &< (W-1)(L_{\max}-1)+\sigma+\rho(a-s) \\ W(t-a-1) &< (W-1)(L_{\max}-1)+\sigma, \end{aligned}$$

The last inequality is due to $\rho \le W$. It can be rewritten as $t < a + L_{\max} + \frac{\sigma - L_{\max}+1}{W}$. This implies $t \le a+f$ because *t* and the time flexibilities are integer.

We will show that $t \le a+f$ implies the theorem. First, note that since the time flexibility of the virtual packet is at least *f*, we have $a+f \le b$. Then $t \le a+f$ implies $t \le b$, and so the constraint $t \in [a,b]$ is satisfied, i.e., the lightpath service begins within the start interval of the request. Finally, we will verify that all requests were assigned with no wrap around. Note that the inequality $t \le a+f$ implies that requests are assigned to lightpath services that start by time $a+f$, and end by time $a+f+L_{\max}-1$. Recall that $A(t)=0$ for the last $f+L_{\max}-1$ time slots in $[0,T-1]$. Therefore, $a \le T-(f+L_{\max})$. Then the lightpath services end by $T-1$. Since none of the lightpath services wrap around, FCFS assigns all requests. ■

The problem with the FCFS assignment is that it tends to assign lightpaths close to time 0, which creates peak bandwidth demand around that time. This leads to higher wavelength requirements. For the proof of Theorem 2, we use another assignment strategy. The strategy first distributes the requests to the wavelengths using a load balancing heuristic. Then for each wavelength, it assigns time slots to requests.

The following lemma is for the case of a single wavelength.

Lemma 3. Suppose $W=1$ and the lightpath requests are (σ, ρ) constrained. In addition, suppose $\sum_{t=0}^{T-1} A(t) \le T$ and $\rho \le 1$. Then there is an assignment for the lightpath requests if their time flexibilities are at least $\lfloor \sigma+\rho-1 \rfloor$.

Proof. We will use the VP Model. We partition the time slots into intervals so that for each interval [*x*,*y*], the first slot *x* has arrivals of virtual packets and the other slots do not. We refer to these as *single-backlog* intervals because they have the following property. Use the FCFS algorithm to assign only the virtual packets of a single-backlog interval [*x*,*y*], starting from time slot *x* and assuming the system is empty. This creates a single *busy period* [*x*,*z*]. If $\|[x,z]\| > \|[x,y]\|$ then we say it *overflows*.

If a single-backlog interval has a busy period that overflows into the next single-backlog interval then we merge the two intervals and apply FCFS 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) and it overflows into itself. Otherwise, its busy period is over T time slots, which contradicts the constraint $\sum_{t=0}^{T-1} A(t) \le T$. Eventually, there will be only single-backlog intervals that do not overflow.

After the mergings of intervals, we consider an arbitrary virtual packet (a,b,L). Let t denote the first time slot when it gets its wavelength. Note that the virtual packet (a,b,L) is part of some single-backlog interval, and let s denote the beginning of this interval. Also note that all virtual packets that arrive in the interval $[s,a]$ are the ones assigned to the time slots in $[s,t]$. Therefore, $\sum_{n\in[s,a]} A(t) \ge \|[s,t]\|$, which implies $\sigma + \rho\|[s,a]\| \ge \|[s,t]\|$. Since $\|[s,t]\| = \|[s,a]\| + \|[a,t]\| - 1$, we have $\sigma - (1-\rho)\|[s,a]\| \ge \|[a,t]\| - 1$. Since $\rho \le 1$ and $\|[s,a]\| \ge 1$, we have $\sigma - (1-\rho) \ge \|[a,t]\| - 1$. Then $t \in [a,b]$ because (*i*) $\|[a,t]\| - 1$ is the delay between a and t and (*ii*) the time flexibilities are at least $\lfloor \sigma + \rho - 1 \rfloor$. Thus, the FCFS assignments are proper and the lemma is proved. ■

The following simple load-balancing 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 $A(t)/W$ of the work. The algorithm keeps track of virtual packets (and their work) assigned to each wavelength, where initially no virtual packets are assigned. The algorithm considers virtual packets in the order they arrive starting from time 0. The virtual packets that arrive at time t are considered on at a time in some arbitrary order and are assigned to the wavelengths with the least work.

The load balancing has the following properties. 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 load balancing algorithm insures 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 note that a wrap around interval $[s,t]$, 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 $[s,t]$,

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

Note that the last inequality is also true for any interval $[s,t]$.

Proof of Theorem 2. Let the load balancing algorithm assign the virtual packets to wavelengths, and 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*. Inequality (2.3) 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) constrained, $\sum_{n\in[s,t]} A_k(n) \le 3L_{\max} + (\sigma + \rho|[s,t]|)/W$. Therefore, the lightpath requests assigned to wavelength *k* are (σ',ρ') constrained, where $\sigma' = 3L_{\max} + \frac{\sigma}{W}$ and $\rho' = \frac{\rho}{W}$. Also note from Inequality (2.1), we have $\sum_{n\in[0,T-1]} A_k(n) \le L_{\max} + \sum_{n\in[0,T-1]} A(n)/W$. Recall that $\sum_{n\in[0,T-1]} A(n) \le W(T - L_{\max})$. Thus, $\sum_{n\in[0,T-1]} A_k(n) \le T$. Now we can apply Lemma 3 to wavelength *k* using the parameters (σ',ρ'). Hence, Theorem 2 is true for wavelength *k*. Since the wavelength is arbitrary, the theorem is true for all wavelengths. ■

3. Assignment Heuristics

We will describe assignment heuristics that perform well for a random ("typical") batch of lightpath requests. Then we will present our simulations of the heuristics and compare their performances.

3.1. Heuristics

<u>First Come First Served (FCFS):</u> This was described in Section 2. The following rules resolve ambiguities of the earlier description and introduces blocking. When FCFS assigns virtual packets to wavelengths, lowest valued wavelengths are used first. Virtual packets with earlier arrival times are assigned first, and ties are broken randomly. If a virtual packet (a,b,L) is unassigned by time $b+L$ then it is *blocked*.

<u>Earliest Deadline First (EDF):</u> This is the same as FCFS except when assigning virtual packets to wavelengths, preference is given to virtual packets (a,b,L) with the earliest *deadline*, i.e., $b+L$, the last possible time slot for the end of the lightpath.

FCFS and EDF tend to assign lightpaths close to time 0 which creates a peak bandwidth demand at time 0. The next heuristics tend to avoid the problem. They fill wavelengths with lightpath requests one wavelength at a time. In this way, they assign lightpath requests in batches, where each batch is evenly distributed over time.

<u>Lowest Wavelength, Maximum Duration (LWMD):</u> Wavelengths are filled with lightpath requests one wavelength at a time and starting from wavelength 0. To fill wavelength *k*, lightpath requests that have longer durations are considered first, where ties are broken randomly. To fit a lightpath request (a,b,L) into the wavelength, start times are considered in the start interval $[a,b]$ beginning with *a*.

Note that preference is given to requests with longer durations because finding assignments for them are more difficult.

Lowest Wavelength, Fixed (LWFixed): Wavelengths are filled with lightpath requests one wavelength at a time starting from wavelength 0. Wavelength k is filled starting from time t = 0 as follows. Choose the longest unassigned request (a,b,L) that could start at time t and assign it starting from t. Then continue filling the wavelength from time $t+L$ (here, the addition is modulo T). If there is no such request then continue filling the wavelength from time t+1. This continues until we reach the end of $[0,T-1]$.

The problem with LWFixed is it always fills wavelengths from time 0. Thus, like FCFS and EDF, it creates peak bandwidth demand at time 0. The next heuristic tends to avoid this.

Lowest Wavelength, Continuous (LWCont): This 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.

3.2. Simulations

We considered two scenarios for simulations.

Blocking Scenario. The number of wavelengths W = 30, and the number of time slots T = 64. There is a batch of 114 lightpath requests, and all have the same time flexibilities f. Their earliest start times are random and uniformly distributed over the time slots. Their durations are random and uniformly distributed over time interval [1,31]. Thus, the expected duration of a lightpath request is 16. The average utilization of the wavelengths is expected to be 95%, though the actual average utilization may be lower due to blocking.

Under this scenario, we measured the next two blocking rates for the assignment heuristics. *Call blocking* is the ratio of the number of blocked lightpath requests over the total number of lightpath requests, which is 114. For this measure, the blocking of a short lightpath request counts the same as the blocking of a long one. The next blocking rate accounts for the durations of the requests. *Traffic blocking* is the ratio of the work of blocked lightpath requests over the work of all lightpath requests.

Nonblocking Scenario. This is similar to the Blocking Scenario with the following differences. First, the number of lightpath requests in a batch is 128 rather than 114. Second, under this scenario, we measure the minimum number of wavelengths so that there is no blocking.

Under each scenario, we randomly generated 100 batches of lightpath requests. Then for each batch, we simulated the heuristic assignments and measured their performances. The performances were averaged over the 100 instances.

Figures 2 and 3 are for the Blocking Scenario. They have the average call and traffic blocking rates, respectively, as a function of the time flexibility f. As expected, FCFS and EDF have high blocking rates. LWCont has about the lowest blocking rates over all flexibility times. Blocking rates decrease as time flexibility increases, except for LWFixed when the time flexibility is around 32. Increased time flexibility allows LWFixed to put more lightpaths at time 0 which degrades its performance.

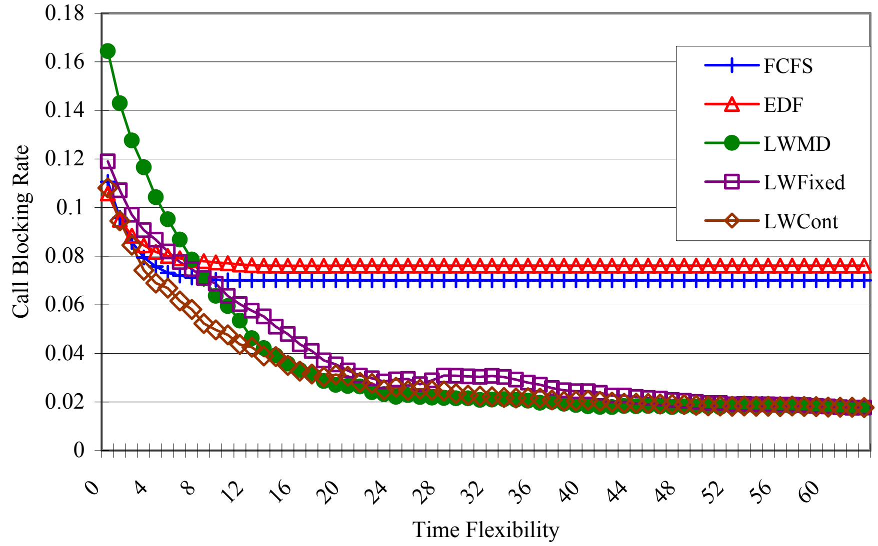

Figure 2. Call blocking rates.

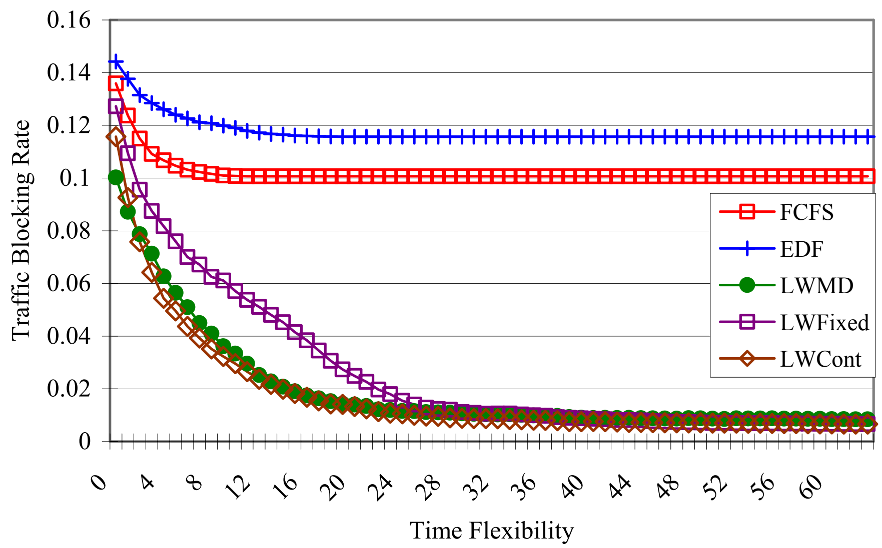

Figure 3. Traffic blocking rates.

Figure 4 shows the simulation results for the Nonblocking Scenario. The performance measure is the average minimum number of wavelengths so there is no blocking. The value *Cmin* is $\lceil M / T \rceil$, where *M* is the work of the lightpath requests. Thus, it is a lower bound on the minimum number of wavelengths. Note that LWMD and LWCont require minimal number of wavelengths when the time flexibility is around 16, the average duration of a lightpath request.

From these simulations, LWCont performs the best over the three performance measures.

4. Conclusions

We considered a single WDM link and the problem of assigning periodic lightpath services, which are allowed some flexibility on when they begin. We applied a traffic characterization by Cruz [1] that shows a relationship between wavelength efficiency, lightpath request burstiness, and time flexibility. We also studied the assignment problem for randomly generated lightpath requests. Simple assignment heuristics were shown to be effective by simulation. Heuristics that filled one wavelength at a time were the most efficient since their assignments are spread over time.

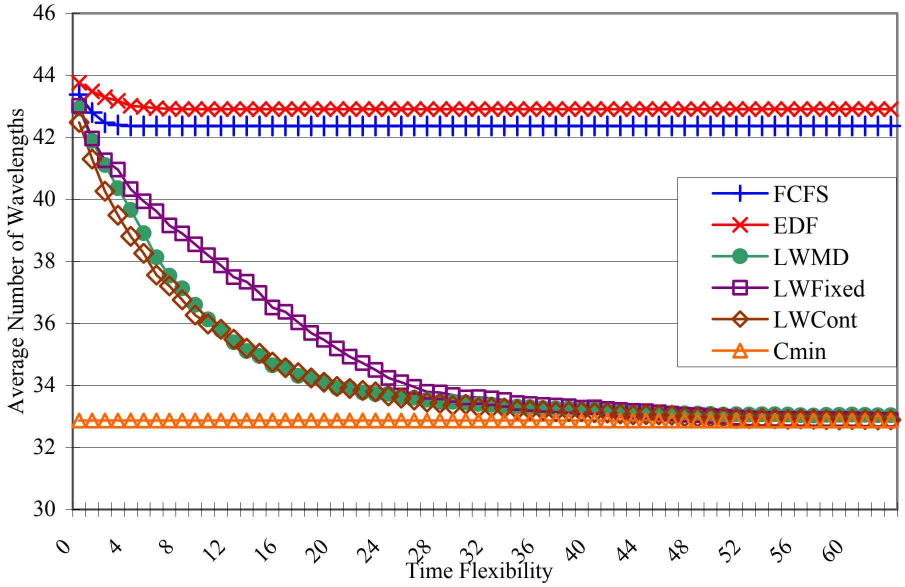

Figure 4. The average number of wavelengths so there is no blocking.

References

1. R. Cruz, "A calculus for network delay, part I: network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114-131, Jan. 1991.
2. O. Gerstel, S. Kutten, R. Ramaswami, and G. Sasaki, "Worst-case analysis of dynamic wavelength allocation in optical networks, *IEEE/ACM Transactions on Networking*, vol. 7, no 6, pp. 833-846, Dec. 1999.
3. R. Ramaswami and G. Sasaki, ``Multiwavelength optical networks with limited wavelength conversion," *IEEE/ACM Trans. Networking*, vol. 6, no. 6, pp. 744-754, Dec. 1998.
4. X. Zhang and C. Qiao, "Scheduling all-to-all personalized connections and cost-effective designs in WDM rings," *IEEE/ACM Trans. Networking*, vol. 7, pp. 435-445, June 1999.