

EFFICIENT SCHEDULING OF PERIODIC TRAFFIC FOR WDM NETWORKS

A THESIS SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAI'I AT MĀNOA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

IN
ELECTRICAL ENGINEERING

MAY 2012

By

John Salle

Thesis Committee:

Galen H. Sasaki, Chairperson

Tep Dobry

Yingfei Dong

We certify we have read this thesis and that, in our opinion, it is satisfactory in scope and in quality as a thesis of the degree of Master of Science in Electrical Engineering.

THESIS COMMITTEE

Chairperson

Acknowledgements

I am very grateful for the guidance and support of my advisor, Dr. Galen H. Sasaki. He generously gave his time and valuable advice, even on short notice, throughout my work. Without his support this thesis would not be possible. I would like to thank my wife Kaitlin and my children, Luke and Logan, for the numerous sacrifices they have made, their unwavering support and encouragement, and their patience throughout my educational pursuit. I also would like to thank Dr. Tep Dobry and Dr. Yingfei Dong for their willingness to work with me, their supportive advice, and for agreeing to serve on my thesis committee. Finally, I would like to thank all of my professors and friends within the Department of Electrical Engineering at the University of Hawaii for their support.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	1
1.1 WDM Technology	1
1.2 WDM Networks and Lightpaths.....	2
1.3 Time Flexibility	4
1.4 Offline and Online Scheduling Algorithms.....	5
1.5 Our Contribution and Related Work.....	6
CHAPTER 2: NETWORK MODEL	8
2.1 The Traffic Scenario	8
CHAPTER 3: SCHEDULING ALGORITHMS	12
3.1 First-Fit.....	12
3.2 Most-Used	14
3.3 First-Fit with Defragmentation.....	17
3.4 Maximum-Product	19
3.5 Maximum-Product with Flexibility	21
CHAPTER 4: SIMULATION RESULTS	23
4.1 Simulation Environment	23
4.2 Simulation Results	27
4.2.1 Uniform Traffic.....	28
4.2.2 Rectangular Traffic	32
4.2.3 Modified Gaussian Traffic	36
4.2.4 Topology Comparison.....	39
4.3 Simulation Conclusion	42
CHAPTER 5: CONCLUSION	43
5.1 Summary	43

5.2 Future Work.....43

REFERENCES44

LIST OF FIGURES

<u>FIGURE</u>	<u>PAGE</u>
1.1 A WDM fiber.....	1
1.2 A WDM network	2
1.3 Traffic load through an OC-192 interface of an Internet data collection monitor.....	4
2.1 Time-Wavelength (TW) representation of a WDM optical fiber link	9
2.2 An example of a feasible assignment across two optical fiber links	11
3.1 An example fiber link scheduling prior to a lightpath request.....	13
3.2 TW matrix of a fiber path. Includes possible matches for a lightpath request.....	13
3.3 Scheduling result of FF given the example lightpath request	14
3.4 Pseudo code for the FF heuristics algorithm	14
3.5 (a) Most-used wavelength for two links. (b) Most-used wavelength for a path.....	16
3.6 The scheduling result of MU	17
3.7 Pseudo code for the MU heuristics algorithm	17
3.8 (a) Scheduled with FF has 3 fragments. (b) Scheduled with FFDe has 2 fragments...	19
4.1 Three example start-time distributions of simulated lightpath requests	26
4.2 (a) The NSFNET mesh topology. (b) The COST 239 mesh topology	27
4.3 Uniform start-time distribution using diverse duration traffic	30

4.4	Uniform start-time distribution using short duration traffic.....	31
4.5	Rectangular start-time distribution using diverse duration traffic	34
4.6	Rectangular start-time distribution using short duration traffic	35
4.7	Modified Gaussian start-time distribution using diverse duration traffic	37
4.8	Modified Gaussian start-time distribution using short duration traffic.....	38
4.9	Five topology comparisons for uniform distribution	40

LIST OF TABLES

<u>TABLE</u>		<u>PAGE</u>
3.1	Example MP calculations for five lightpath requests	20
3.2	A sorted list of lightpath requests based on MP scheduling difficulty	21
3.3	A sorted list of lightpath requests based on MPflex scheduling difficulty	22
4.1	Performance Summary of traffic with variable flexibility	42

CHAPTER 1

INTRODUCTION

1.1 WDM Technology

With the realization of Wavelength Division Multiplexing (WDM), Internet Service Providers have been able to provide higher bandwidth across optical fiber communication links. WDM technology allows for multiple wavelengths within an optical fiber to function as separate data channels [3]. These channels are then multiplexed together onto a single fiber for long distance transmission [4]. Figure 1.1 gives a graphical representation of this concept. Each of the three wavelengths in the diagram are initiated from a different data source and then multiplexed together across a single optical fiber. The technology has come so far since its invention in 1970 [3] [4] that today's implementation is able to carry data rates of 101.7 terabits/s across a single optical fiber link over 165km utilizing 370 separate channels [8].

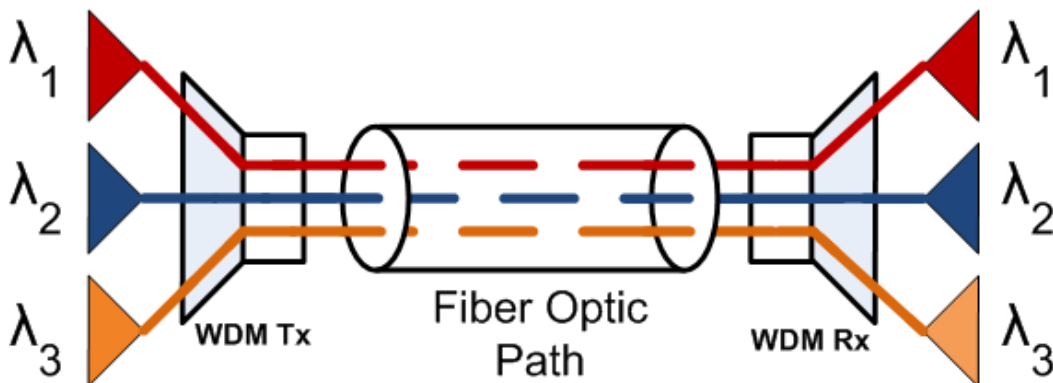


Figure 1.1. A WDM fiber.

1.2 WDM Networks and Lightpaths

A WDM network is composed of WDM switches connected together via optical fiber cabling as shown in Figure 1.2. A WDM switch can switch signals on different wavelengths on incoming ports to different outgoing ports. Access nodes connect to WDM switches and for the purposes of this paper we will refer to a WDM switch and access node together as WDM nodes, or simply nodes. An optical end-to-end communications connection between two nodes spanning multiple physical fiber links using a single wavelength is known as a *lightpath* [1] [2] [7] [6] [5] [4] [3]. We will assume that when a lightpath passes through a WDM switch it cannot change wavelengths. This is true for WDM switches that switch signals using only optical components. Notice that a lightpath will use the same wavelength along its entire route. This constraint is known as the *wavelength continuity constraint* [1]. Another constraint is that lightpaths traversing an optical fiber link at the same time must use different wavelengths.

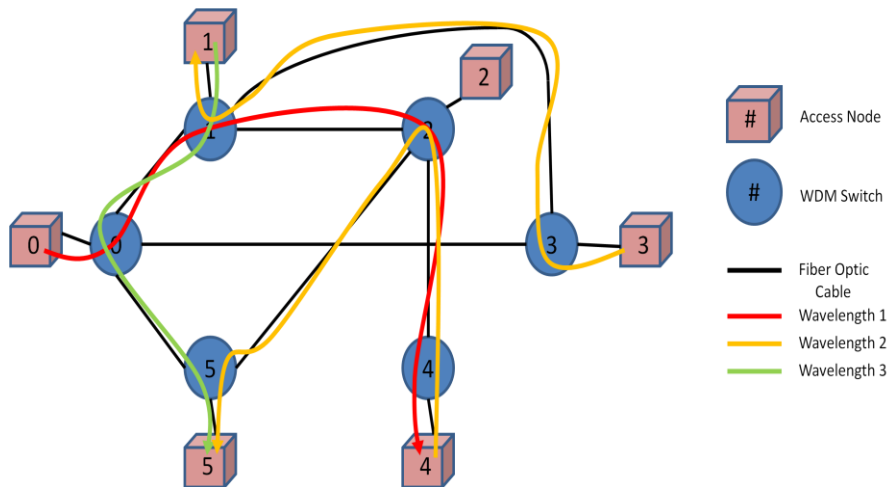


Figure 1.2 A WDM network.

The problem of setting up a lightpath requires computing a physical route and reserving an open wavelength. This is known as the *Routing and Wavelength Assignment* (RWA) Problem which has been proven to be NP-complete [1]. To make it more tractable, it can be broken up into two sub-problems: finding a route, and then assignment of a wavelength [1] [3] [4]. A set of heuristics can be used to solve the routing sub-problem and then another set of heuristics can be used to solve the wavelength assignment sub-problem.

Lightpath request traffic in a WDM network can be modeled into two distinct types: *static* and *dynamic traffic* [1]. In a static traffic model, all lightpath requests are known ahead of time, and once a lightpath has been established, its assignment and route cannot change. An optimization problem with respect to this model is to schedule all the lightpath requests in a manner that utilizes the least amount of resources (i.e. wavelengths). In a dynamic traffic model, lightpath requests have random arrival times, and therefore each request is scheduled one at a time, as opposed to the global approach taken with the static traffic model. For dynamic traffic the goal is to minimize the number of blocked requests and maximize the number of connections that can be established at any given moment.

In both traffic models a lightpath request that contains both a connection setup and teardown time is known as a *scheduled lightpath* [2]. These requests are a type of advanced reservation of a lightpath [15] [16]. The assignment of a scheduled lightpath request to an available wavelength is known as *scheduling* [2].

Advanced reservations of lightpaths are typical given certain end user applications, such as company system backups or scientific experimentation setup between two remote sites [5] [6] [7]. Often, these types of connection requests are not meant for one-time use, but periodic use such as daily or weekly lightpath reservations [2]. We denote such lightpaths as *periodic*

lightpaths [2]. It has been shown that traffic patterns are periodic across WDM networks [3] [4]. A clear illustration of traffic periodicity can be seen in Figure 1.3, which shows the traffic load over the course of a week through an OC-192 interface of an Internet data collection monitor in San Jose, CA [11].

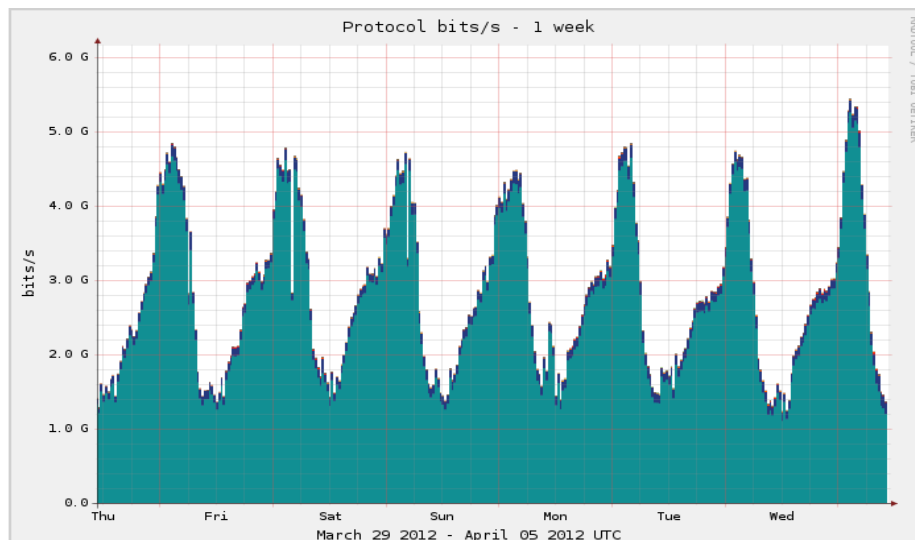


Figure 1.3. Traffic load through an OC-192 interface of an Internet data collection monitor in San Jose, CA [11].

If portions of the traffic load could be shifted to non-peak hours then Figure 1.3 would present a more uniform traffic distribution. Ultimately, a lower and more efficient wavelength utilization would result. The ability of a customer to alter the times in which they request network resources is known as *time flexibility* [2], and it is discussed further in the following section.

1.3 Time Flexibility

Time flexibility was introduced in [2] as a useful attribute of scheduled lightpath requests. It was further analyzed in [3] [4] [13] [14]. However, in [13] [14] lightpath requests that have time flexibility are referred to as *sliding scheduled traffic requests*. Initially, lightpath requests

contained precise setup and teardown times for network service. However, not all applications require such stringent service windows. For example, in today's cloud architectures, system replications need to take place on a regular basis. A company may require one hour of guaranteed lightpath service to complete their server replications on a daily basis. However, the data transfers may be accomplished at any time during the hours of 1am and 5am when their user base is minimal. Therefore, we say the customer is flexible with respect to the start-time of their lightpath service. Additionally, presenting a service provider with a time-range for when a lightpath reservation is needed may offer benefits for both the customer and the ISP. Commercial service providers have Time-of-Day pricing tiers that charge more for lightpath durations which fall within peak traffic hours [2]. Offering the service provider a flexible start-time allows for optimizations to be made within the ISP's network resources by scheduling the lightpath during lower traffic periods. This, in turn may lead to reduced pricing for the customer. The service provider in turn can optimize the use of its resources.

1.4 Offline and Online Scheduling Algorithms

There are two different classes of *lightpath scheduling algorithms* used to solve the scheduling sub-problem depending on whether the traffic model is static or dynamic [12]. One class, known as *offline algorithms*, uses the static traffic model where a collection of lightpath requests are known ahead of time. The algorithm then schedules all of the lightpath requests together so as to utilize the minimal amount of wavelengths. The other class, known as *online algorithms* utilizes the dynamic traffic model and schedules lightpath requests one at a time, as they arrive. In this thesis, the discussion will focus on *offline algorithms*. However, in the

conclusion chapter, we discuss how these offline algorithm results lead to implications about online algorithms.

1.5 Our Contribution and Related Work

Numerous papers have looked at the problem of scheduling [1] [2] [3] [4] [5] [6] [7]. The relationship that time flexibility shares with periodic scheduled lightpaths and the number of wavelengths used was first proposed and analyzed in [2]. The work in [2] focused on a single WDM fiber link connecting two nodes. Scheduling algorithms were proposed and analyzed by simulation. Performance was based on the number of wavelengths utilized assuming static traffic. To simplify analysis, for each simulation batch of lightpath requests, the flexibility values were identical. The results of [2] show that with a modest amount of time flexibility, wavelength usage is considerably diminished. In [3] a dynamic traffic model was used and five online algorithms were proposed and analyzed using the same topology and traffic distributions in [2]. It was shown through simulation that online algorithms displayed comparable performance to offline algorithms. In [4] routing and scheduling lightpaths through a ring topology was considered. The routes of the lightpaths were shortest-hop paths, and lightpaths for the same source-destination pairs use the same paths. Additionally in [4], their online algorithms were used to do offline scheduling. In particular, a preprocessing algorithm that sorts a batch of lightpath requests was presented. Then the online algorithms would schedule the requests in that order. It was shown that intelligent ordering improved the wavelength utilization.

In this thesis we follow the work done in [3] [4] by analyzing algorithm performance using the same scheduled traffic model with flexibility. However, our contribution will be to analyze performance across a mesh topology whereas [3] and [4] were restricted to single links

or rings. Additionally, a more realistic traffic simulation is conducted in which flexibility values for lightpath requests can vary within a simulation batch. This is different from the analysis in [2] [3] [4] where simulations were conducted using identical flexibility values for all lightpath requests in a given simulation batch. Finally, we propose a new scheduling algorithm to take into account traffic sets composed of varying flexibilities. We present simulations that show our algorithm performs significantly better than existing online algorithms.

This thesis is organized as follows. In Chapter 2 we describe the WDM topologies and traffic model. In Chapter 3 we discuss the top four best-performing scheduling algorithms from [3] [4], as well as our newly proposed algorithm. Chapter 4 presents our simulation results and compares the algorithms' performance to one another across different traffic distributions. Additionally, we look at the effect the topology has on the algorithm's performance. Finally, in Chapter 5 conclusions will be made and a discussion of future work will be given.

CHAPTER 2

NETWORK MODEL

In this section we introduce the traffic model and its associated terminology used for our analysis. Our analysis will focus on mesh topologies, such as the one displayed in Figure 1.2. We assume that lightpaths are routed using shortest-hop paths. Thus, the remaining problem is the wavelength and timeslot assignment problem.

2.1 The Traffic Scenario

We assume scheduled lightpaths occur periodically once every day. We also organize time into a collection of T evenly spaced timeslots, labeled $[0, 1, 2, \dots, T-2, T-1]$, and together they form one period. The time period can also apply to any arbitrary period such as a week or a month but for the sake of discussion we assume it's a day. In our case, each timeslot has a duration of 10 minutes, therefore $T = 144$. Additionally, we define timeslot 0 and timeslot $T/2$ as midnight and noon respectively.

Figure 2.1 illustrates a single WDM fiber link, where rows represent wavelengths and columns correspond to timeslots. We refer to this as the Time-Wavelength (TW) matrix for the link.

Using the interval $[0, T-1]$ as our period, we define the following subsets:

- An *ordinary interval* $[s, t]$ is simply a subinterval of $[0, T-1]$, where $s \leq t$.
- If $s, t \in [0, T-1]$ and $s > t$, then a *wrap-around interval* exists and can be written as $[t, t+1, \dots, T-1, T, 0, \dots, s]$.
- The magnitude of an interval $[s, t]$ is denoted as $|[s, t]|$.

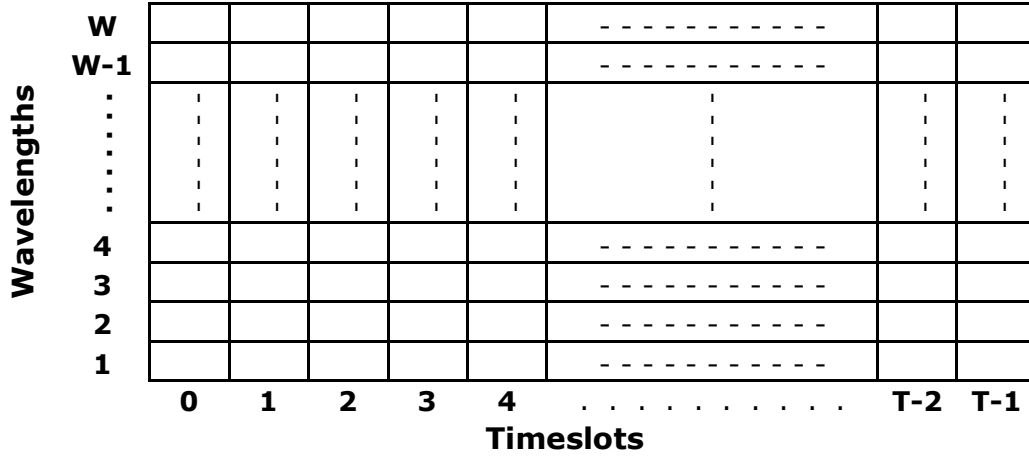


Figure 2.1. Time-Wavelength (TW) representation of a WDM optical fiber link.

Our mesh topology graph consists of N nodes and L links. Each optical fiber link is denoted by $l_{(x,y)}$, where $x,y \in [0, N-1]$ identifies the link by its end nodes x and y .

A *lightpath request* is used to make a reservation of a lightpath service and is represented by a 5-tuple (src, dst, a, b, D) :

- The (src, dst) pair represents the lightpath's source and destination nodes respectively.
- The values $a,b \in [0, T-1]$ denote the earliest timeslot a and the latest timeslot b service can begin. We refer to the interval $[a, b]$ as the *start window*.
- The value D represents the duration of the lightpath in timeslots and is known as the *service duration*.

The *start window* can be either an *ordinary interval*, or a *wrap-around interval*. We use this interval to define the lightpath request's *time flexibility* $f = |[a, b]|-1$. For the case where a and b are the same there is no flexibility. Conversely, no restriction implies $|[a, b]| = T$.

A *scheduled lightpath* that satisfies a lightpath request (src, dst, a, b, D) must:

- Have a path from src to dst .

- Begin transmission in a timeslot on the interval $[a, b]$.
- Have service duration D .
- Must be assigned a wavelength w that is unused by any other lightpath on any of its link during its duration.

If we assign lightpath requests that satisfy the *scheduled lightpath* requirements, we say we have a *feasible assignment*. An example of a *feasible assignment* is shown in Figure 2.2. In the example we present three lightpaths scheduled over the optical fiber links $l_{(0,1)}$ and $l_{(1,2)}$. The period T is composed of 15 timeslots labeled $[0, 1, \dots, 13, 14]$. Lightpath request $(0, 2, 5, 7, 5)$ is assigned to wavelength 1 on fiber links $l_{(0,1)}$ and $l_{(1,2)}$ as required by the *wavelength continuity constraint*. Request $(0, 2, 0, 0, 6)$ is assigned to wavelength 2 on fiber links $l_{(0,1)}$ and $l_{(1,2)}$ and also conforms to the *wavelength continuity constraint*. Notice request $(0, 2, 0, 0, 6)$ has no flexibility since $|[0, 0]|-1 = 0$. The third request $(0, 1, 6, 1, 10)$ is assigned to wavelength 3, but only on fiber link $l_{(0,1)}$. This is because the (src, dst) pair is $(0, 1)$ and therefore only needs to traverse fiber link $l_{(0,1)}$. Notice request $(0, 1, 6, 1, 10)$ is a *wrap-around interval*. Additionally, all three lightpaths share multiple timeslots, but they are each assigned to different wavelengths. Since all *scheduled lightpath* requirements are met, we have a feasible assignment. We refer to a feasible assignment of scheduled lightpaths as a *schedule* [2].

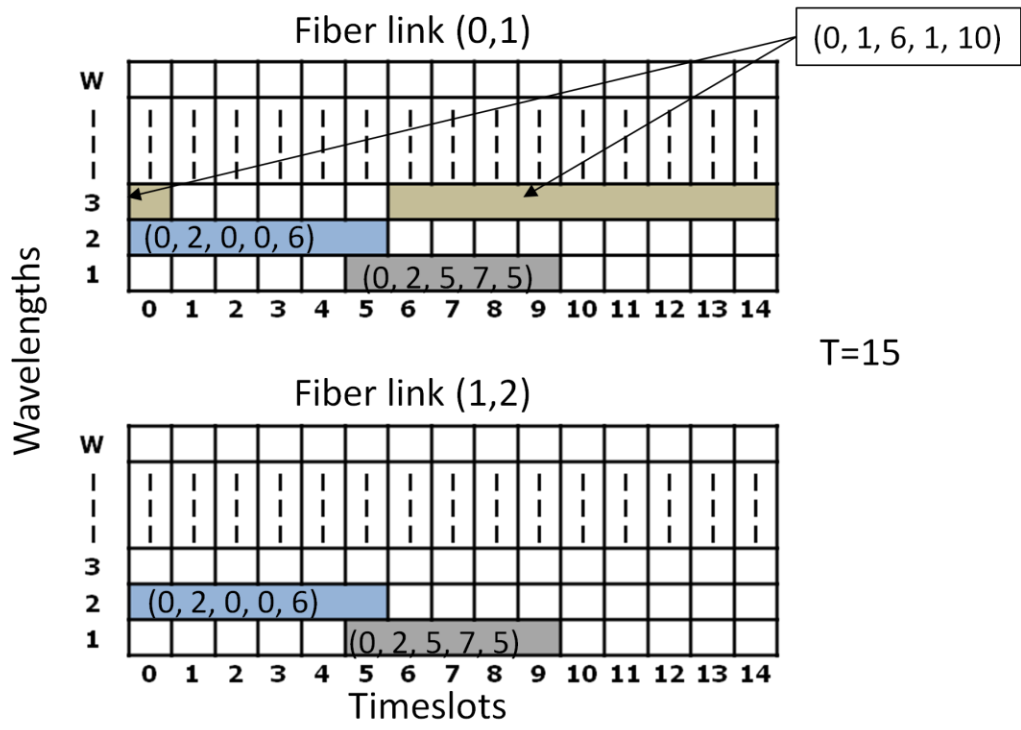


Figure 2.2. An example of a feasible assignment across two optical fiber links. Three lightpath requests have been assigned that satisfy all scheduled lightpath requirements.

CHAPTER 3

SCHEDULING ALGORITHMS

This section first reviews the top four scheduling algorithms from works in [3] [4]. Three of the algorithms, First-Fit, Most-Used, and First-Fit with Defragmentation assume there is a randomly ordered list of lightpath requests. The algorithms schedule the requests one at a time in the order of the list. However, the order in which the requests appear in the list can affect performance. So the fourth algorithm, the Maximum-Product algorithm, orders the lightpath requests to improve performance. Additionally, our new algorithm also orders requests to improve performance. However, it uses the *time flexibility* of the requests to determine the order. This will be presented in Subsection 3.5.

3.1 First-Fit Algorithm

The First-Fit (FF) heuristics algorithm [3] [4] attempts to assign lightpaths to the lowest numbered wavelength which has an available timeslot. This has a packing characteristic such that lightpath requests tend to be scheduled to the lower numbered wavelengths. Additionally, the FF algorithm tries to assign a lightpath request to its earliest service duration start-time.

As an example of how the algorithm works, consider the lightpath request ($\text{src} = 1, \text{dst} = 4, a = 0, b = 7, D = 3$). Suppose the route of the lightpath is $l_{(1,2)}$ and $l_{(2,4)}$. Figure 3.1 shows the time-wavelength (TW) matrices for the two links, where the white cells are unused, and the dark cells are occupied by other lightpaths. We can superimpose the TW matrices of the links to form a TW matrix of the path as shown in Figure 3.2, where the white cells are unused on the entire path, and the dark cells are occupied.

Wavelength 2 will be selected given it is the lowest wavelength with the ability to service the request. However, there are two possible matches and therefore the request is reserved using the earlier start-time. The final assignment is shown in Figure 3.3. Pseudo code for the First-Fit heuristics algorithm is given in Figure 3.4.

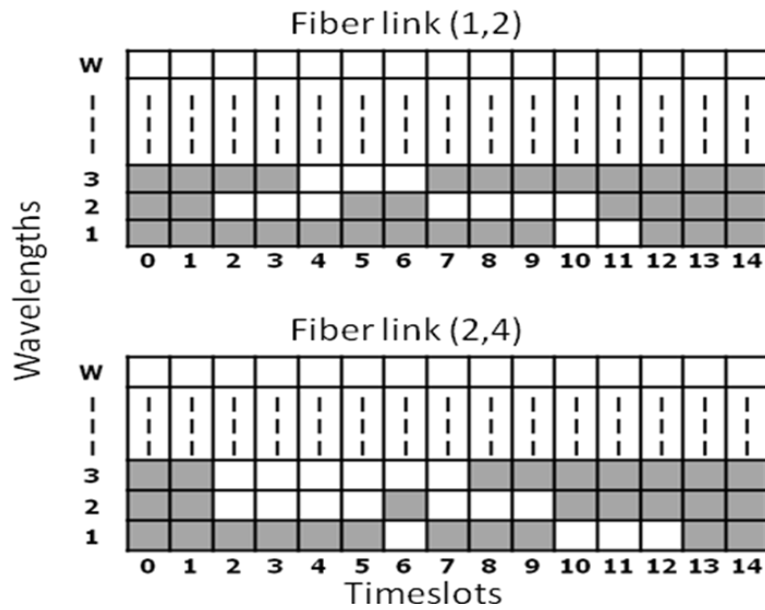


Figure 3.1. An example fiber link scheduling prior to lightpath request (1, 4, 0, 7, 3).

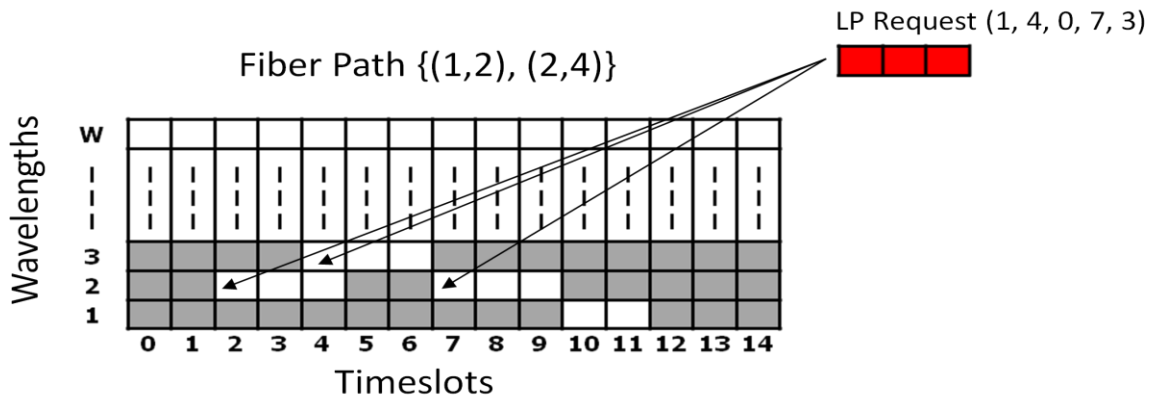


Figure 3.2. TW matrix of fiber path $\{l_{(1,2)}, l_{(2,4)}\}$. Includes possible matches given lightpath request (1, 4, 0, 7, 3).

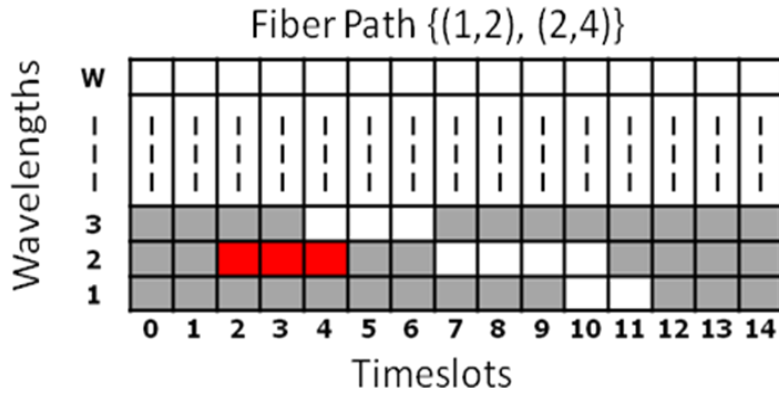


Figure 3.3. Scheduling result of FF given the example lightpath request (1, 4, 0, 7, 3).

FF Algorithm:

Input: A lightpath request with source node **src**, destination node **dst**, earliest start-time **a**, latest start-time **b**, service duration **D**, and a path **p**.

begin

create a TW matrix for path **p** by superimposing the TW matrices of its links ;

$w = 1$;

while the lightpath is not scheduled

```

for  $t = a$  to  $b$ 
    use the TW matrix of p to determine if the lightpath request can be
    scheduled in wavelength  $w$  starting in time  $t$  ;
    if it can then stop ;
 $w = w + 1$  ;

```

end

Figure 3.4. Pseudo code for the FF heuristics algorithm.

3.2 Most-Used Algorithm

The Most-Used (MU) heuristics algorithm [3] [4] monitors the timeslot utilization or traffic load on each wavelength along every fiber link for a given lightpath. The decision of which wavelength to assign a lightpath request is done by choosing the available wavelength

with the highest traffic load [3] [4]. This allows wavelengths to be utilized more efficiently by trying to fill a wavelength up before using a new wavelength. Similar to the FF algorithm, MU will choose the earliest service duration start-time if multiple options exist. We illustrate the MU method with the following example. Consider the same lightpath request $(1, 4, 0, 7, 3)$ and route $\{l_{(1,2)}, l_{(2,4)}\}$ as in the FF example in Subsection 3.1. Figure 3.5(a) shows the TW matrices for each link on the path, similar to Figure 3.1. The figure also shows the number of occupied timeslots or cells per wavelength. For example, link $l_{(1,2)}$ has 13 used cells on wavelength 1, 8 used cells on wavelength 2, and 12 used cells on wavelength 3.

Figure 3.5(b) shows the TW matrix of the path. Here, wavelength 1 has 24 used cells because there are 13 used cells on link $l_{(1,2)}$ and 11 used cells on link $l_{(2,4)}$.

The MU algorithm would consider using the most used wavelengths first for a scheduled lightpath. In the case of wavelengths 1, 2, and 3, the algorithm would first consider wavelength 1, then 3, and finally 2. Since a lightpath cannot be scheduled in wavelength 1, the algorithm chooses wavelength 3 as shown in Figure 3.6. Pseudo code for the MU heuristics algorithm is given in figure 3.7.

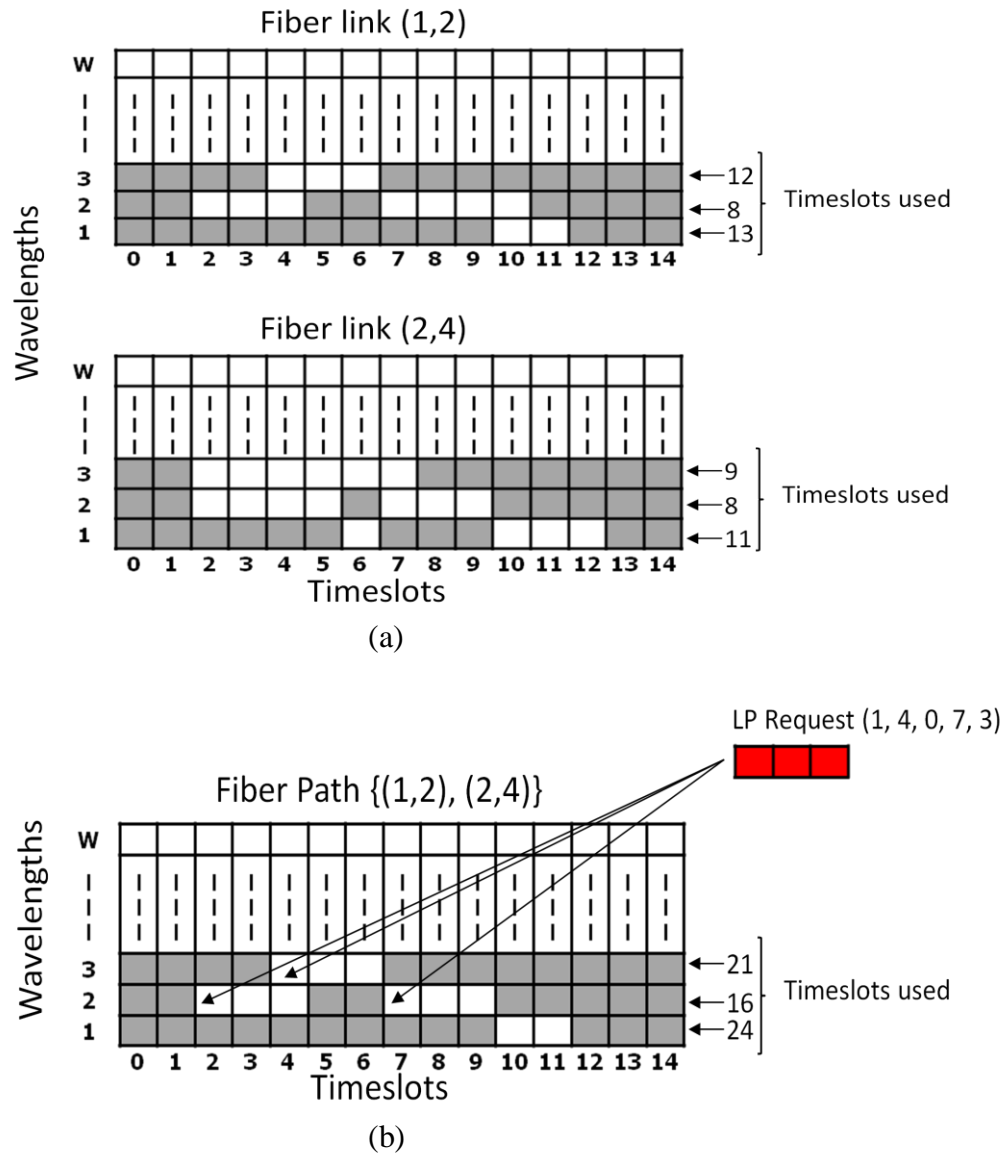


Figure 3.5. (a) Calculation of most-used wavelength for links $l_{(1,2)}$ and $l_{(2,4)}$. (b) Calculation of most-used wavelength for the path $\{l_{(1,2)}, l_{(2,4)}\}$. Includes three possible assignments for request (1, 4, 0, 7, 3).

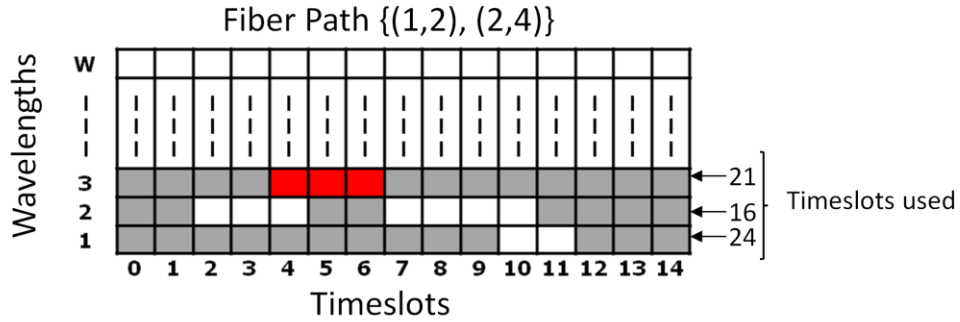


Figure 3.6. The scheduling result of MU.

MU Algorithm:

Input: A lightpath request with source node **src**, destination node **dst**, earliest start-time **a**, latest start-time **b**, service duration **D**, and a path **p**.

begin

for each TW matrix along the path **p**, compute the number of used cells per wavelength ;

create a TW matrix for path **p** by superimposing the TW matrices of its links. Also sum the number of used cells per wavelength along the path ;

sort wavelengths from the ones with the most used cells to the ones with the least used cells ;

for each wavelength **w** in sorted order

if the lightpath can be scheduled in wavelength **w** according to the TW matrix of the path **p**, then schedule it using the earliest start-time **a**, and then stop.

end

Figure 3.7. Pseudo code for the MU heuristics algorithm.

3.3 First-Fit with Defragmentation Algorithm

When a wavelength is chosen by the FF heuristic it may create gaps of small duration that make future scheduling along the same wavelength hard to accomplish. We refer to these gaps as fragments and an example can be seen in Figure 3.8(a). In this example a lightpath with a

service duration of 3 timeslots (in red) has been scheduled across 2 fiber links starting at timeslot 4. By the nature of the FF heuristic, even if the lightpath request had flexibility to start at timeslot 5, it would not, because the earlier timeslot 4 is available. As you can see from Figure 3.8(a), fiber link $l_{(1,2)}$ has 1 fragment and fiber link $l_{(2,4)}$ has 2 fragments. The First-Fit with Defragmentation (FFDe) heuristic [3] [4] attempts to minimize the fragmentations produced as a result of the FF algorithm by utilizing the time flexibility offered in a lightpath request. FFDe monitors the total number of fragmentations created by a given reservation and then attempts to shift the service in time to the right to reduce the number of fragments. At each time shift, FFDe recalculates the number of fragments generated and if the previous fragment count is larger than or equal to the current count, the time duration is shifted again as long as there is flexibility to do so. Ultimately, the scheduling with the smallest fragment count is chosen, and if two schedules tie, the scheduling with the earliest start-time is chosen. Figure 3.8(b) shows the scheduling FFDe would have chosen instead of the lightpath schedule chosen by the FF heuristic in Figure 3.8(a). You can see in Figure 3.8(b) the FFDe scheduling has only 2 fragments as opposed to 3 from the original FF method.

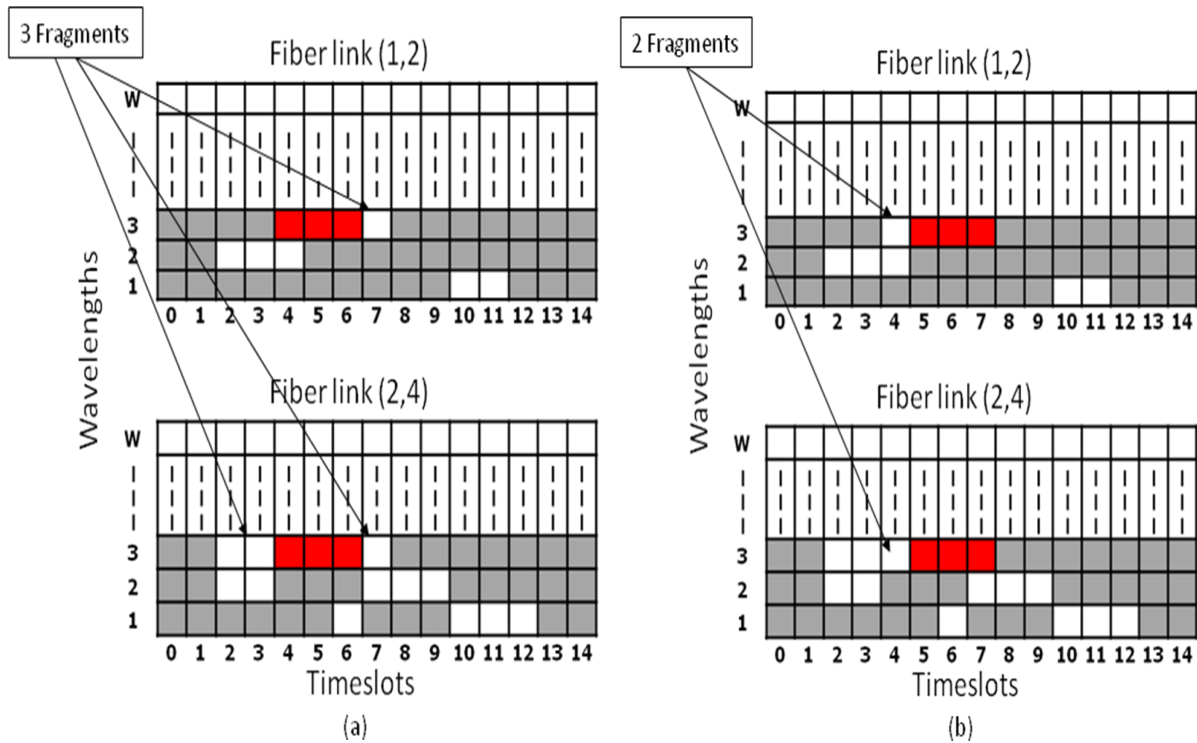


Figure 3.8. Figure (a) shows a request scheduled with FF and contains 3 fragments. Figure (b) shows the same request scheduled with FFDe and only has 2 fragments.

3.4 Maximum-Product Algorithm

The Maximum-Product (MP) [4] heuristic orders the lightpath requests so that the FF, MU, and FFDe algorithms perform better. It orders requests based on *scheduling difficulty*. The assumption is lightpaths that traverse many fiber links or lightpaths that have long service durations will be harder to schedule than lightpaths with a smaller number of hops or service duration. The *scheduling difficulty* is measured as the product of two ratios, namely the *service duration ratio* (SDR) and the *number of fiber links ratio* (NFLR). Lightpath requests are ordered with large scheduling difficulty first. After the ordering is complete, algorithms such as FF, MU, and FFDe can schedule the lightpath requests.

The ratios are constructed as follows: SDR is equal to a requests' service duration D divided by the number of timeslots T for a given period. For example, in Subsections 3.1, 3.2, and 3.3, we consider $T = 15$. Given a requested service duration of 3, the corresponding SDR would be $3/15 = 0.2$. The NFLR is the ratio of length of a lightpath route over the longest shortest-hop length in the network. To compute the longest shortest-hop length, shortest-paths are computed between all pairs of nodes. Their lengths are also computed. Then the longest of these lengths is the longest shortest-hop. For example, in Figure 1.2 the possible shortest-hop lengths are 1, 2, and 3. The longest shortest-hop is 3.

To further illustrate the MP heuristic, Table 3.1 provides an example list of requests ordered by arrival time, and their calculated scheduling difficulty (i.e. product) composed of the SDR and NFLR ratios. The SDR ratio is based on a period $T = 15$, and the NFLR ratio is based on a topology with a longest shortest-hop of 3.

Table 3.1. Example MP calculations for five lightpath requests indexed based on request arrival time.

Lightpath Request Index (based on arrival time)	Service Duration (SD)	Number of Fiber Links (NFL)	SDR (SD/15)	NFLR (NFL/3)	Scheduling Difficulty (SDR*NFLR)
1	6	1	0.40	0.33	0.132
2	5	2	0.33	0.67	0.221
3	4	3	0.27	1.00	0.270
4	3	1	0.20	0.33	0.066
5	2	3	0.13	1.00	0.130

Sorting the requests from Table 3.1 based on the scheduling difficulty is presented in Table 3.2. As you can see from the results, lightpath request index 3 is the most difficult to schedule, and consequently will be scheduled first using an algorithm such as FFDe.

Table 3.2. A sorted list of lightpath requests based on scheduling difficulty. Request index 3 will be scheduled first.

Lightpath Request Index	Service Duration (SD)	Number of Fiber Links (NFL)	SDR (SD/15)	NFLR (NFL/3)	SORTED Scheduling Difficulty (SDR*NFLR)
3	4	3	0.27	1.00	0.270
2	5	2	0.33	0.67	0.221
1	6	1	0.40	0.33	0.132
5	2	3	0.13	1.00	0.130
4	3	1	0.20	0.33	0.066

3.5 Maximum-Product with Flexibility Algorithm

The MP heuristic defines scheduling difficulty based on the length of a requests' *service duration*, and the number of hops between the source and destination of a request [4]. It does not take into account the flexibility of a lightpaths request. Intuitively, one can see as the flexibility of a request decreases, the difficulty to schedule the request will increase. A more concrete explanation can be given by considering to schedule a lightpath request (src, dst, a , b , D) of duration $D = 1$ in a particular wavelength w . Suppose each timeslot on the wavelength w is occupied with probability $p = 1/2$, and the likelihood of occupation is statistically independent among the timeslots. Since the lightpath requires one timeslot, it cannot be scheduled if all timeslots in $[a, b]$ are occupied. This occurs with probability $p^{|[a, b]|}$. The larger the flexibility $f = |[a, b]|-1$, the lower the probability of being blocked from wavelength w . This justification is the premise for developing the Maximum-Product with Flexibility (MPFlex) heuristics algorithm.

The MPFlex heuristic modifies *scheduling difficulty* by including time flexibility of a lightpath request. The scheduling difficulty is measured by the product of three ratios: SDR, NFLR, and a *flexibility ratio* (FR) of a lightpath request. The FR of a lightpath request is defined as $MaxFlex/Flex$ where:

- *MaxFlex* is the maximum flexibility for an entire set of requests plus one.
- *Flex* is the flexibility f of the request plus one.

The motivation for the “plus one” in the definition of *MaxFlex* and *Flex* is to avoid a divide by zero error.

Once scheduling difficulty for all requests have been computed, the requests are then sorted based on their difficulty. Lightpath reservations are then made starting with the largest difficulty request. Using the same set of lightpath requests in Table 3.1 the MPFlex algorithm is implemented and in Table 3.3 the sorted list is displayed. Notice using the additional FR value to compute scheduling difficulty places the requests in a different order than the MP results listed in Table 3.2.

Table 3.3. A sorted list of lightpath requests based on MPFlex scheduling difficulty. Request index 1 will be scheduled first.

Lightpath Request Index (based on arrival time)	Service Duration (SD)	Number of Fiber Links (NFL)	Time Flexibility (f)	SDR (SD/15)	NFLR (NFL/3)	FR ($(7/f+1)$)	SORTED Scheduling Difficulty (SDR*NFLR*FR)
1	6	1	1	0.40	0.33	3.50	0.462
3	4	3	6	0.27	1.00	1.00	0.270
2	5	2	5	0.33	0.67	1.17	0.259
5	2	3	4	0.13	1.00	1.40	0.182
4	3	1	3	0.20	0.33	1.75	0.116

CHAPTER 4

SIMULATION RESULTS

4.1 Simulation Environment

The simulator was written in C language and compiled using the gcc compiler. All simulations were conducted on a MacBook Pro running the Mac OS X 10.7 operating system.

We evaluate the scheduling algorithms described in Chapter 3 based on the number of wavelengths used. An algorithm is said to have better performance if it uses a smaller number of wavelengths to schedule the same set of lightpath requests [2]. Our simulations correspond to the situation where blocking is unacceptable and therefore a service provider would ensure they provision a sufficient amount of wavelengths during capacity planning.

As mentioned in Subsection 2.1 we chose each time slot t to be 10 minutes in duration, and since our traffic is periodic once a day, $T = 144$. For test data we randomly generated sets of lightpath requests with each set containing a fixed number of requests R . Each request consisted of a source and destination node pair that was chosen randomly and uniformly over the interval $[0, N-1]$ assuming each pair of nodes is equally likely. Recall that lightpaths use the shortest-path to be routed over. If there are multiple shortest-paths, only one of them is chosen and used for the duration of the simulations.

In our simulations, time flexibility f was chosen randomly, independently, and uniformly over an interval $[0, f-1]$ for each batch of requests. The *service durations* for each request were also chosen in the same way except over the interval $[1, \delta-1]$, where δ is an input parameter specified during traffic-set generation. This gives an average duration $D_{avg} = \delta/2$ for each batch

of requests. Additionally the expected average traffic load per timeslot is then $\frac{R(\delta/2)}{T}$. Let $\{a_1, a_2, \dots, a_R\}$ denote the earliest start-time values for the lightpath requests of a data set. These values were also chosen randomly and independently over the interval $[0, T-1]$. However, three distinct statistical distributions of $\{a_1, a_2, \dots, a_R\}$ were considered, namely *uniform*, *rectangular*, and *modified Gaussian*. These distributions were proposed in [2] and the motivation behind each type was to mimic different traffic situations that could arise in real WDM networks.

The uniform traffic model distributes start-times over the values $\{a_1, a_2, \dots, a_R\}$ uniformly across the interval $[0, T-1]$. Therefore, the probability density function is $p(t) = \frac{1}{T}$. Using the rectangular model, traffic requests peak during the interval $[T/3, 2T/3]$ and are minimal otherwise. The density function for this model is the following:

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

The modified Gaussian distribution model uses a mean $\mu = T/2$ which implies our peak traffic load should happen around noon each day. The standard deviation is $\sigma = T/6$ (i.e. 4 hours). Note this follows closely with the actual traffic distribution presented in Figure 1.3. The density function is defined by: $p(t) = g(t)/G$, $t \in [0, T-1]$ where $g(t) = \exp\left[-\frac{(t+0.5-\mu)^2}{2\sigma^2}\right]$, and G is the normalization constant $G = \sum_{t=0}^{T-1} g(t)$.

In our simulations there are two traffic duration types: *diverse duration* and *short duration*. The diverse duration type was generated with the parameters $R = 504$, $\delta = 24$, while the short duration traffic was generated with the parameters $R = 2016$, $\delta = 6$. We defined the *lightpath request load* of a timeslot t to be the sum of all the lightpath requests which have an

earliest start-time t . Figure 4.1 illustrates the lightpath request load over the time interval $[0, T-1]$ for three sample batches of traffic using each of the traffic distribution models. The statistical start-time distribution defined as $p(t) \cdot R$ is also plotted for each model.

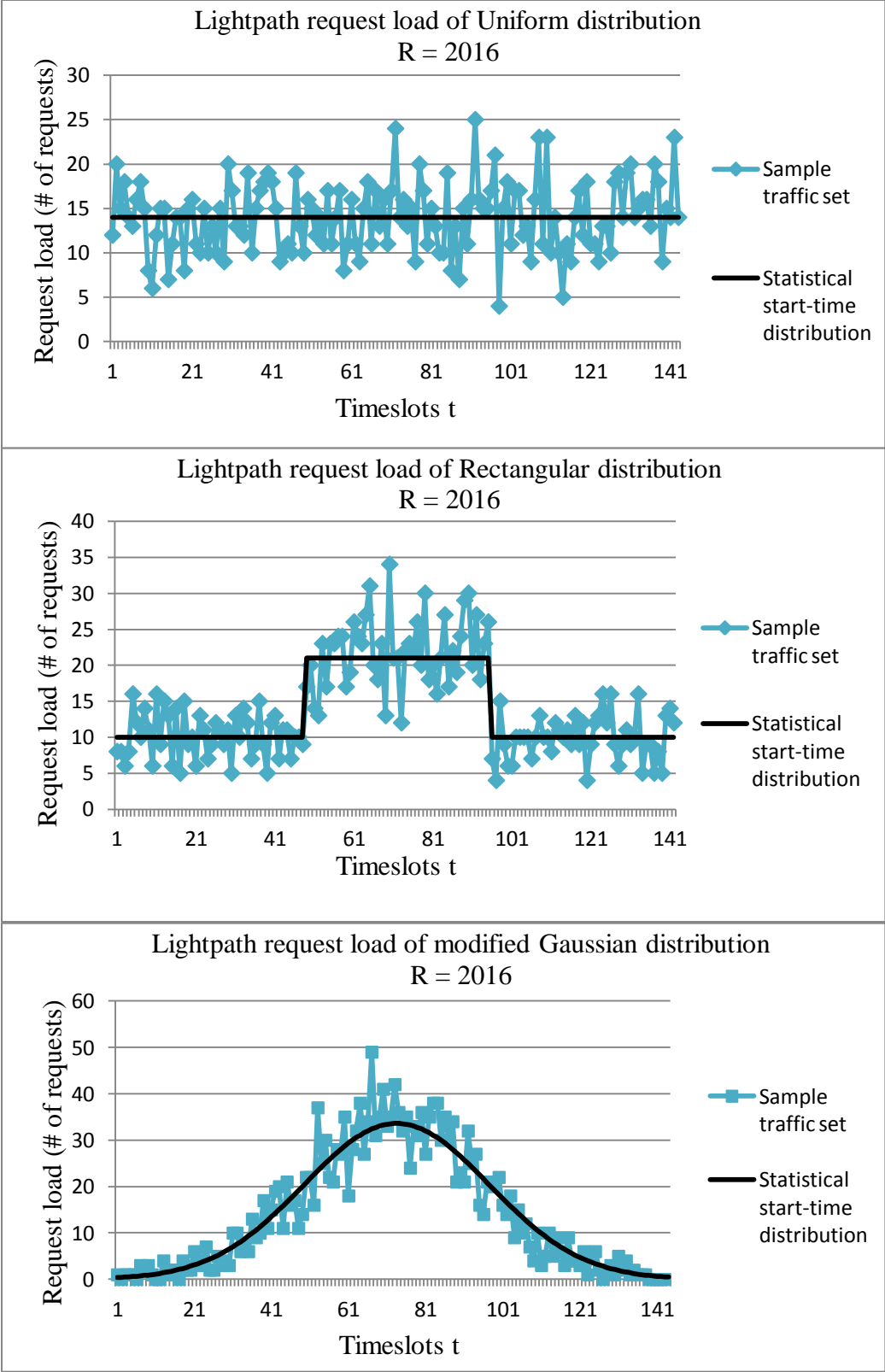


Figure 4.1. Three example start-time distributions of simulated lightpath requests.

4.2 Simulation Results

We ran simulations to evaluate the performance of the various algorithms and how time flexibility and the network topology can affect wavelength utilization. We used four different topologies: a single bidirectional link as in [2] [3], a 6-node bidirectional ring topology as in [4], the National Science Foundation Network (NSFNET) which consists of 14 nodes and 21 links [9], and the Pan-European optical network from the COST 239 project [10] which consists of 11 nodes and 26 links. Figure 4.2 displays the NSFNET and COST 239 topologies.

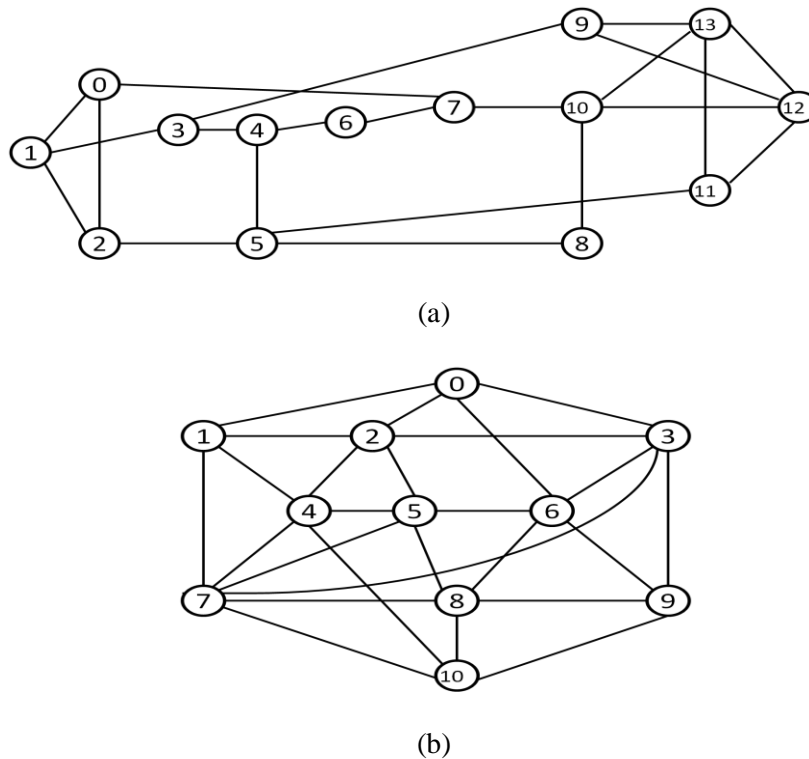


Figure 4.2. The NSFNET optical mesh topology (a) and the COST 239 optical mesh topology (b).

We tested each algorithm using all three traffic distributions on each topology for both short and diverse duration traffic-sets. For each simulation, an algorithm schedules one batch of traffic requests. Each point in the resulting graph is the average of 100 batches. Since there are 14 points on each graph, a total of 1,400 different traffic sets were scheduled by each algorithm

for each graph. We plotted the results as the number of wavelengths used based on the flexibility of the data sets. Recall from the previous section that flexibility varied randomly over an interval $[0, f-1]$ for each batch of traffic requests. In order to graph the results we define each point f on the horizontal axis as the flexibility interval $[0, f-1]$ for which the set of requests varied. For example, in Figure 4.3 the point $f = 6$ means each of the 100 batches tested contained requests with flexibilities randomly varying over the interval $[0, 5]$.

We simulated five scheduling algorithms:

- FF: This is First-Fit where the lightpath requests are randomly ordered.
- MU: This is Most-Used where the lightpath requests are randomly ordered.
- FFDe: This is First-Fit with Defragmentation where the lightpath requests are randomly ordered.
- MPFFDe: This is First-Fit with Defragmentation where lightpath requests are ordered using the Maximum-Product heuristic.
- MPFlexFFDe: This is First-Fit with Defragmentation where lightpath requests are ordered using the Maximum-Product with Flexibility heuristic.

We chose the first four because they performed the best from the work done in [3] [4].

Simulation results comparing the algorithms for different traffic on mesh topologies are presented in Subsections 4.2.1, 4.2.2, and 4.2.3. A comparison over different topologies is given in Subsection 4.2.4.

4.2.1 Uniform Traffic

Figure 4.3 shows the average number of wavelengths used to schedule lightpath requests over the NSFNET and COST 239 topologies for uniform traffic. The lightpath request traffic

contains diverse duration (i.e. $R = 504$, $\delta = 24$). We can see that FF, MU, and FFDe perform the same, where MU is generally slightly worse. MPFFDe has the next best performance as it should since prior to wavelength assignment it orders the requests based on scheduling difficulty (in terms of hop count and service duration). Finally, we see that MPFlexFFDe performs the best. Notice that for diverse duration traffic as the flexibility variance grows, the performance gap between MPFlexFFDe and the other algorithms gets larger. This is because MPFlexFFDe has a wider range of scheduling difficulty ratios to sort as the flexibility variance grows. Because it takes flexibility variance into account (while the others do not), MPFlexFFDe can order the requests more efficiently than the others.

Figure 4.4 evaluates the same scenario as Figure 4.3 but the traffic duration is short (i.e. $R = 2016$, $\delta = 6$) rather than diverse. In this figure the performance gap does not really get larger as the flexibility variance grows. This is due to the little impact service duration has on the scheduling difficulty ratio, since all requests have relatively small service duration $[0, 5]$. At any rate taking flexibility into account for scheduling as MPFlexFFDe does, still yields the best results.

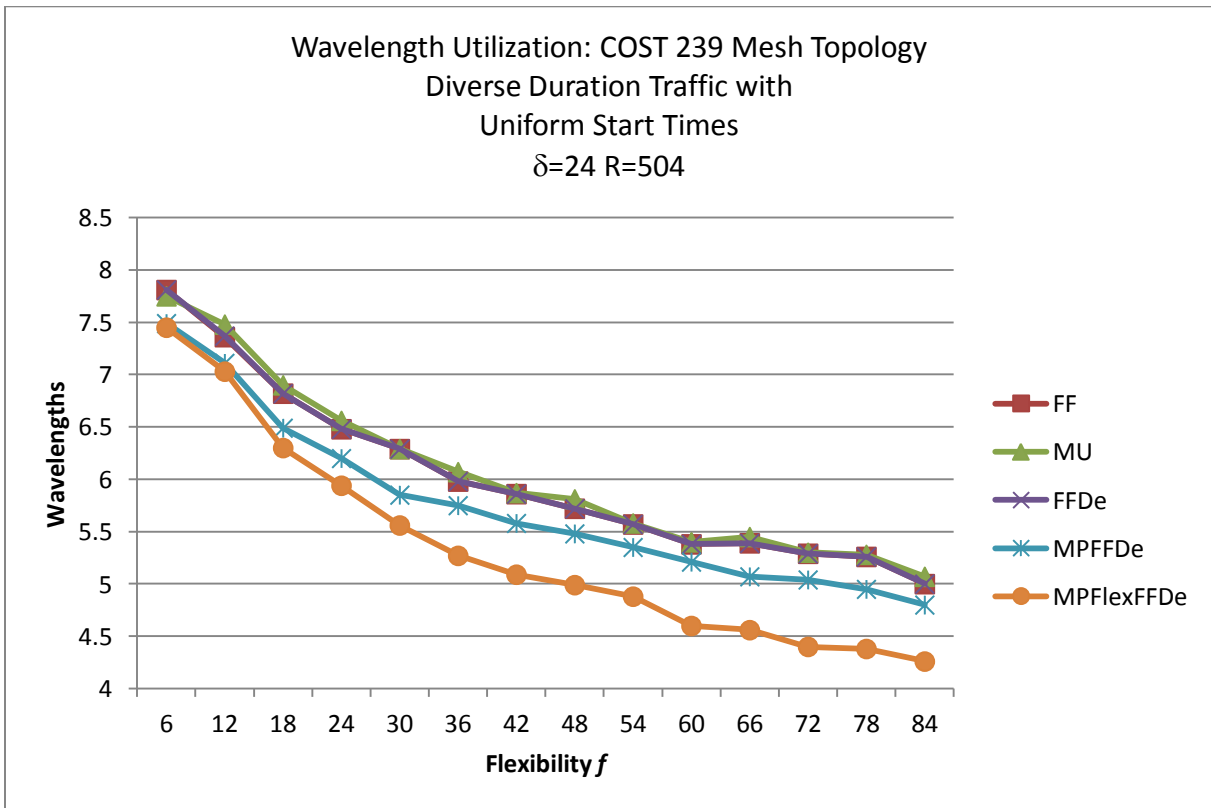
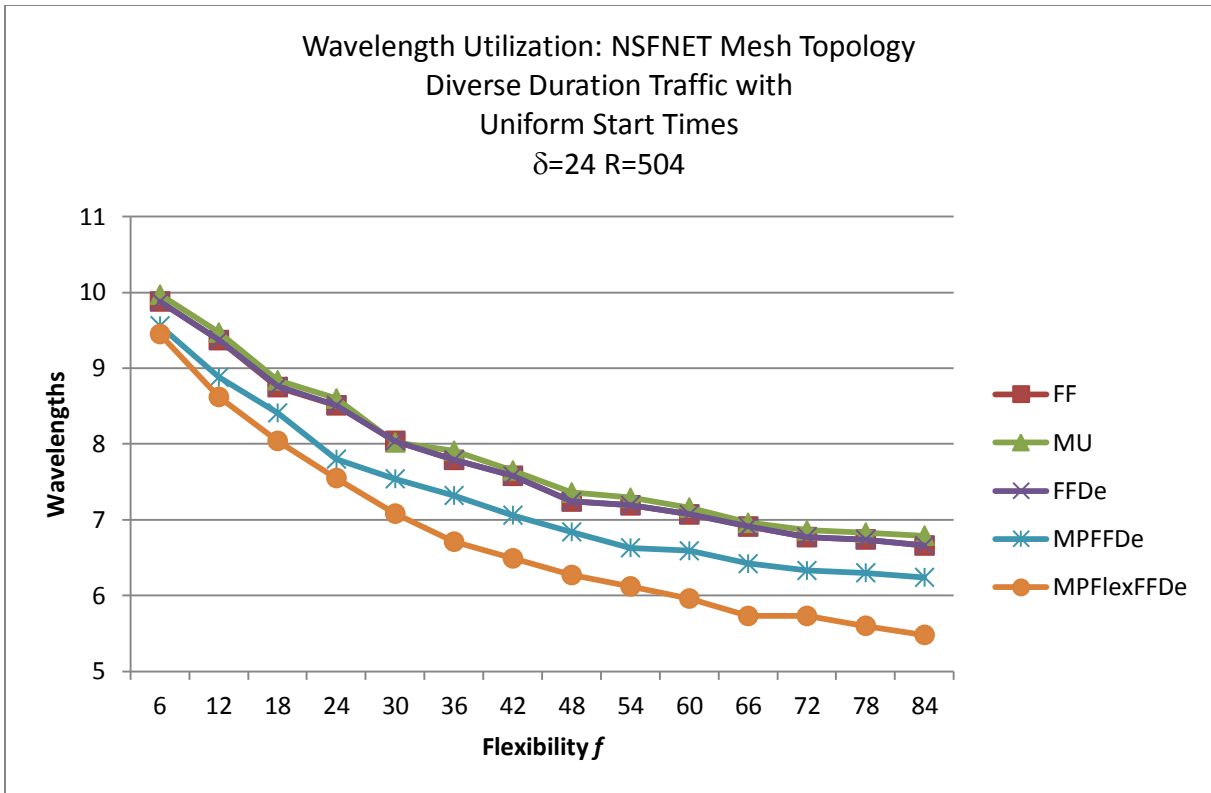


Figure 4.3. Uniform start-time distribution using diverse duration traffic.

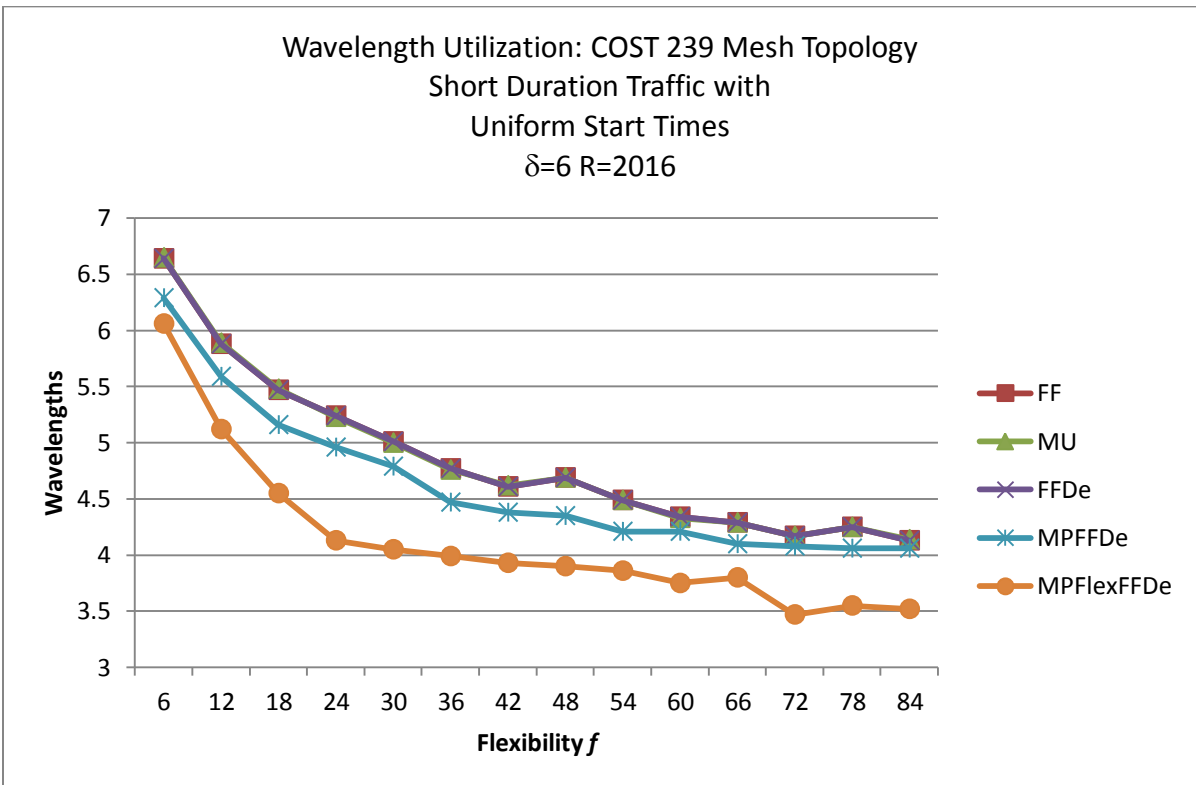
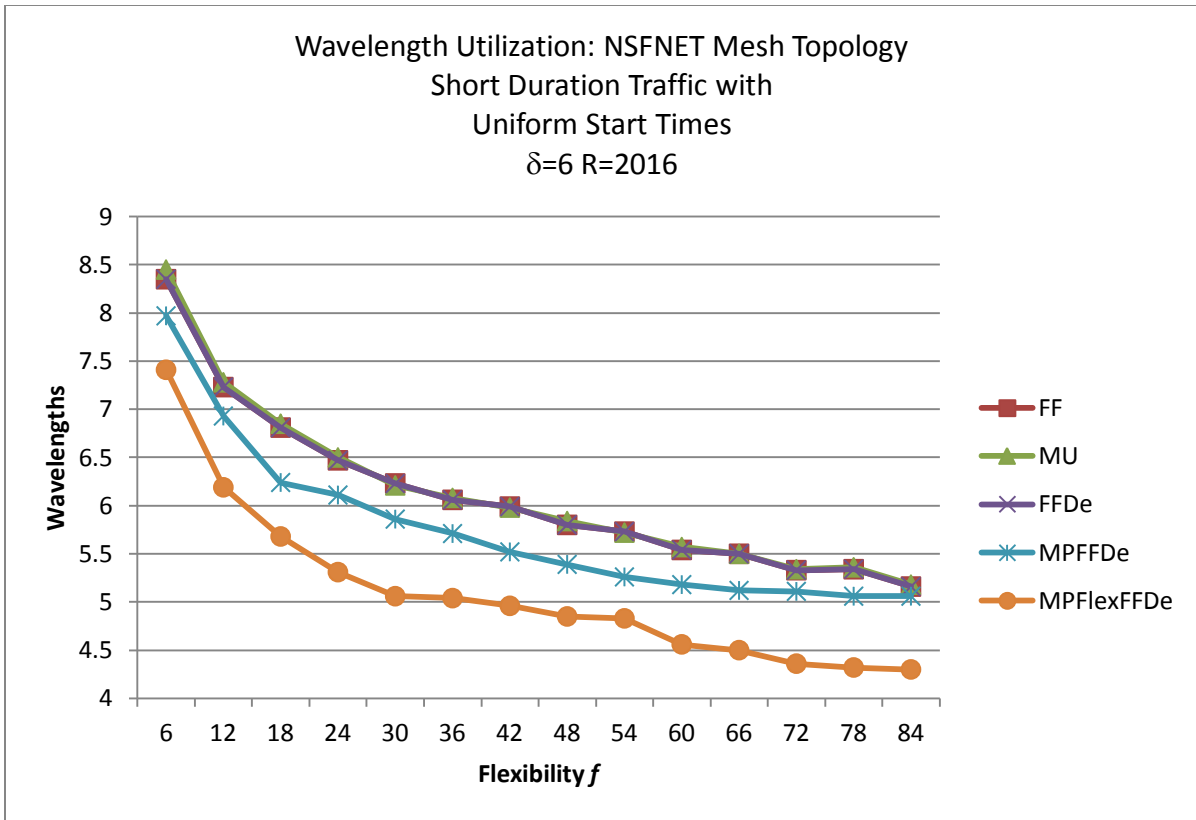


Figure 4.4. Uniform start-time distribution using short duration traffic.

4.2.2 Rectangular Traffic

Figure 4.5 also shows the average number of wavelengths used to schedule lightpath requests over the NSFNET and COST 239 topologies for rectangular traffic. In this figure the lightpath request traffic uses diverse duration (i.e. $R = 504$, $\delta = 24$). We see again that FF, MU, and FFDe perform the same, MPFFDe performs better, and MPFlexFFDe is the most efficient. We again see that for diverse duration traffic, the performance gap between MPFlexFFDe and the other algorithms tends to widen as the flexibility variance grows. This again is due to the same reasons mentioned in Subsection 4.2.1. Notice for diverse duration traffic when the flexibility variance is small both MPFFDe and MPFlexFFDe perform the same. This is because flexibility has little impact on scheduling difficulty and therefore the algorithms are essentially the same.

Figure 4.6 evaluates the same scenario as Figure 4.5, however, the traffic duration is short (i.e. $R = 2016$, $\delta = 6$) rather than diverse. The gap between MPFlexFFDe and the others, again, tends to stay the same. However, notice even for small flexibility MPFlexFFDe is more efficient than MPFFDE. This is because in this scenario flexibility variance has a larger impact on scheduling difficulty than service duration. This tells us when a set of requests all have relatively small service durations, time flexibility is more important in determining the scheduling difficulty of a request.

Each of the algorithms perform essentially the same (compared to each other) in both the uniform distribution case and the rectangular case. However, the rectangular traffic model always uses more wavelengths than the uniform model for the same number of requests. This is due to the rectangular model containing a peak traffic load on the interval $[T/3, 2T/3]$. Another contributing factor is the diverse flexibilities across the requests for each simulation batch. As

flexibility variance increases, more lightpath start-times can be shifted outside the peak interval to present a more uniform shape, but they cannot reach the same uniform distribution unless all lightpath requests contain large flexibilities. In [3] [4] using fixed flexibility for all lightpath requests in a batch, the algorithms were able to mimic a uniform traffic distribution for a large enough flexibility.

Further comparing uniform and rectangular traffic models we see in both cases short duration traffic tends to use fewer wavelengths than diverse duration traffic. This is because short duration traffic tends to fit better into the wavelengths, especially if there is flexibility. This tells us the chances of service duration overlap between lightpath requests can be better handled when all requests have smaller durations.

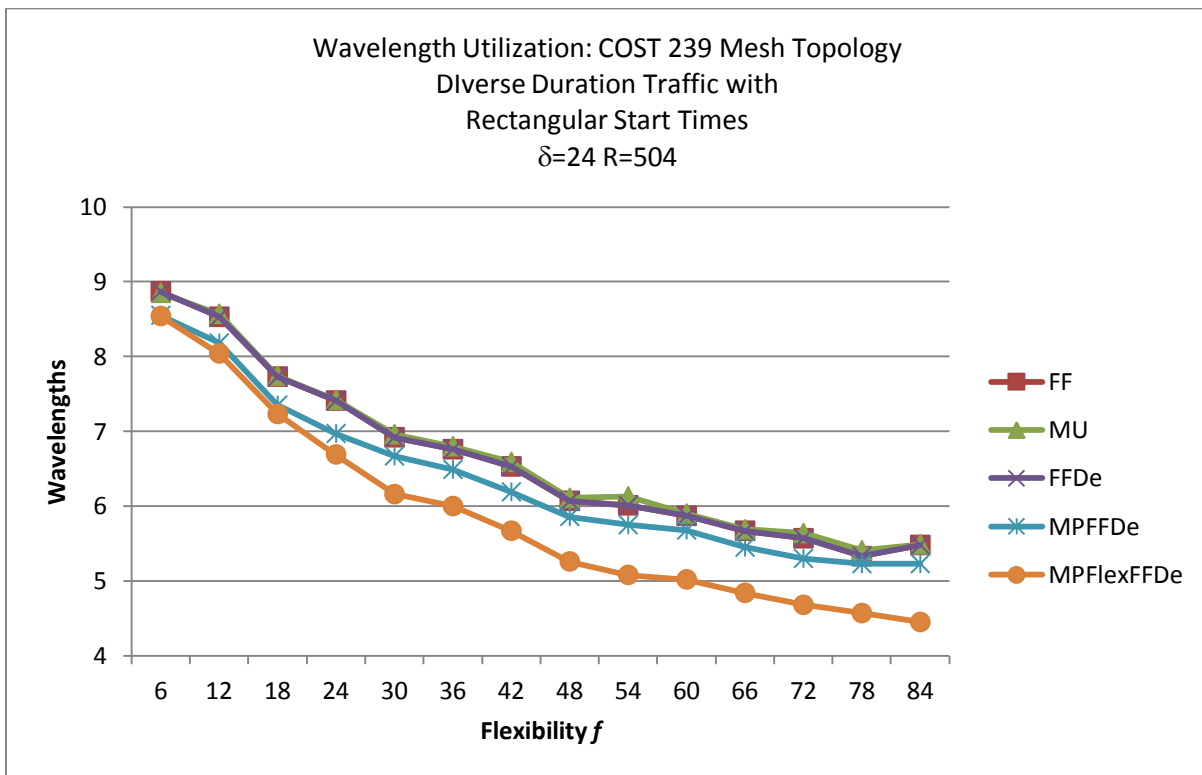
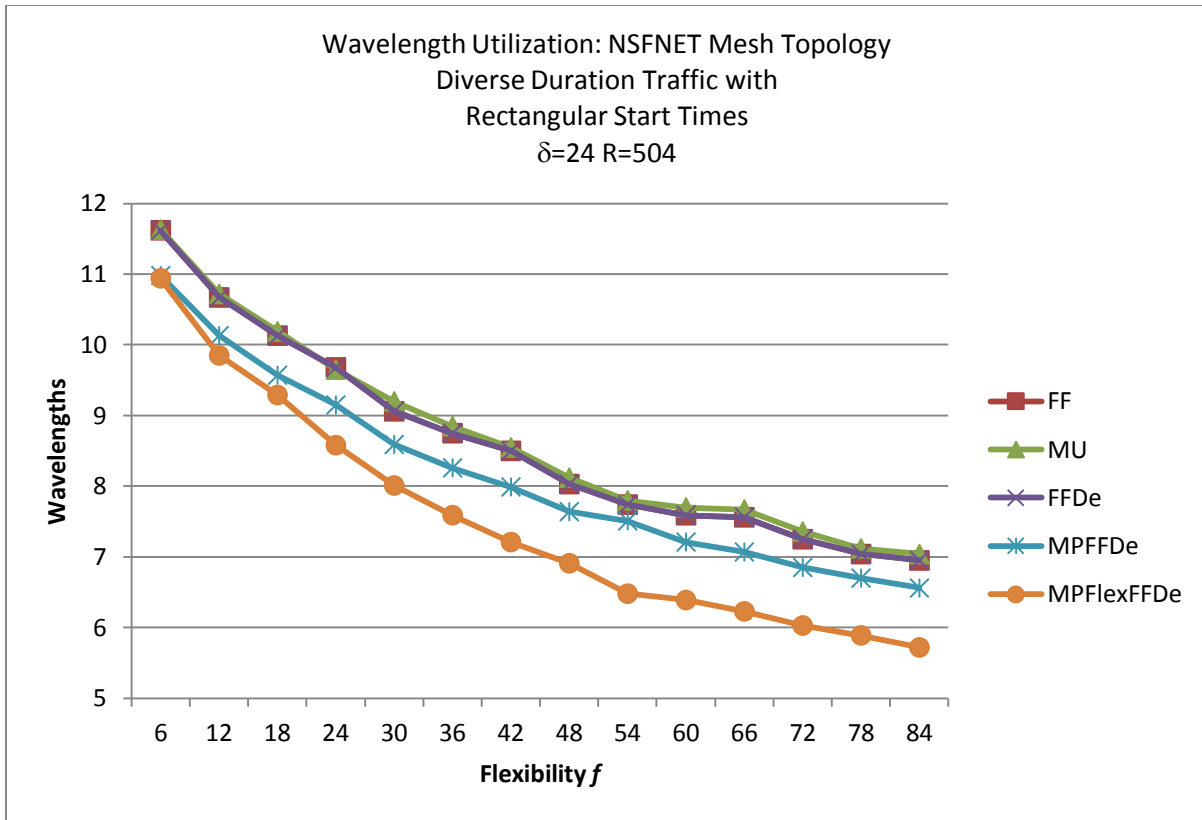


Figure 4.5. Rectangular start-time distribution using diverse duration traffic.

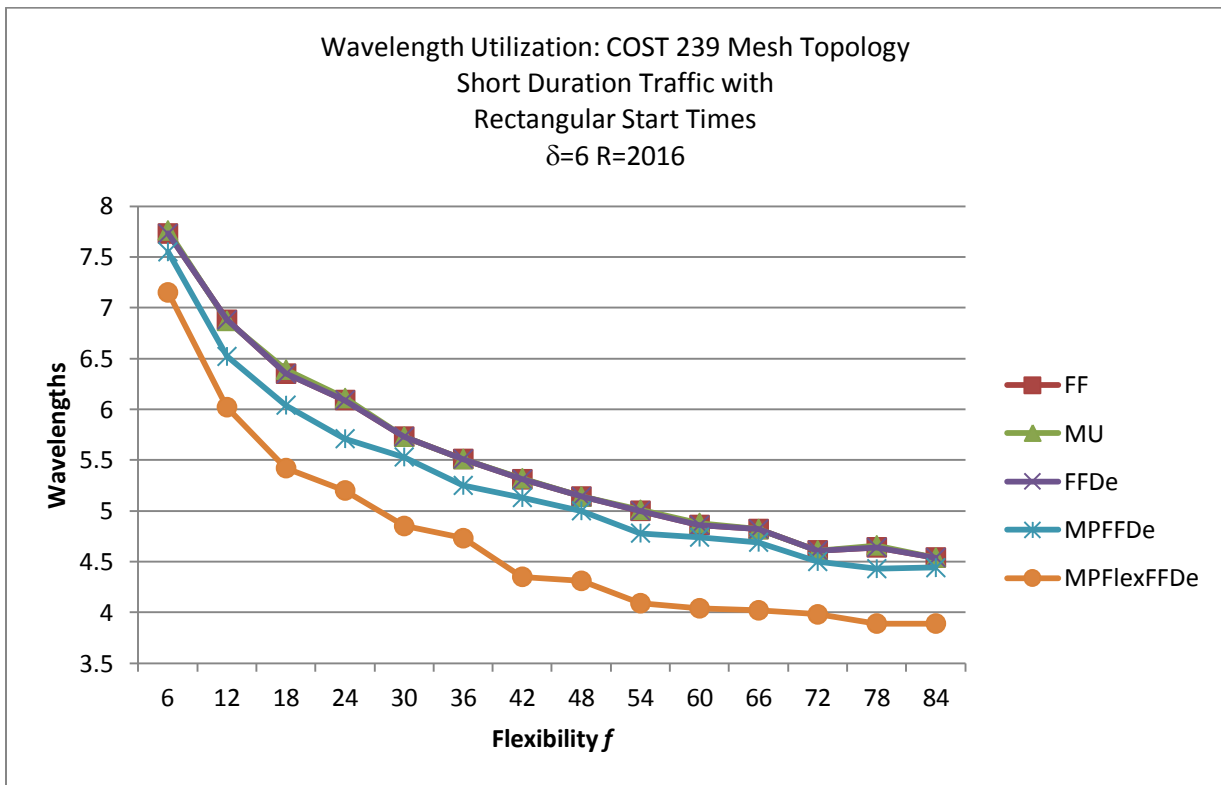
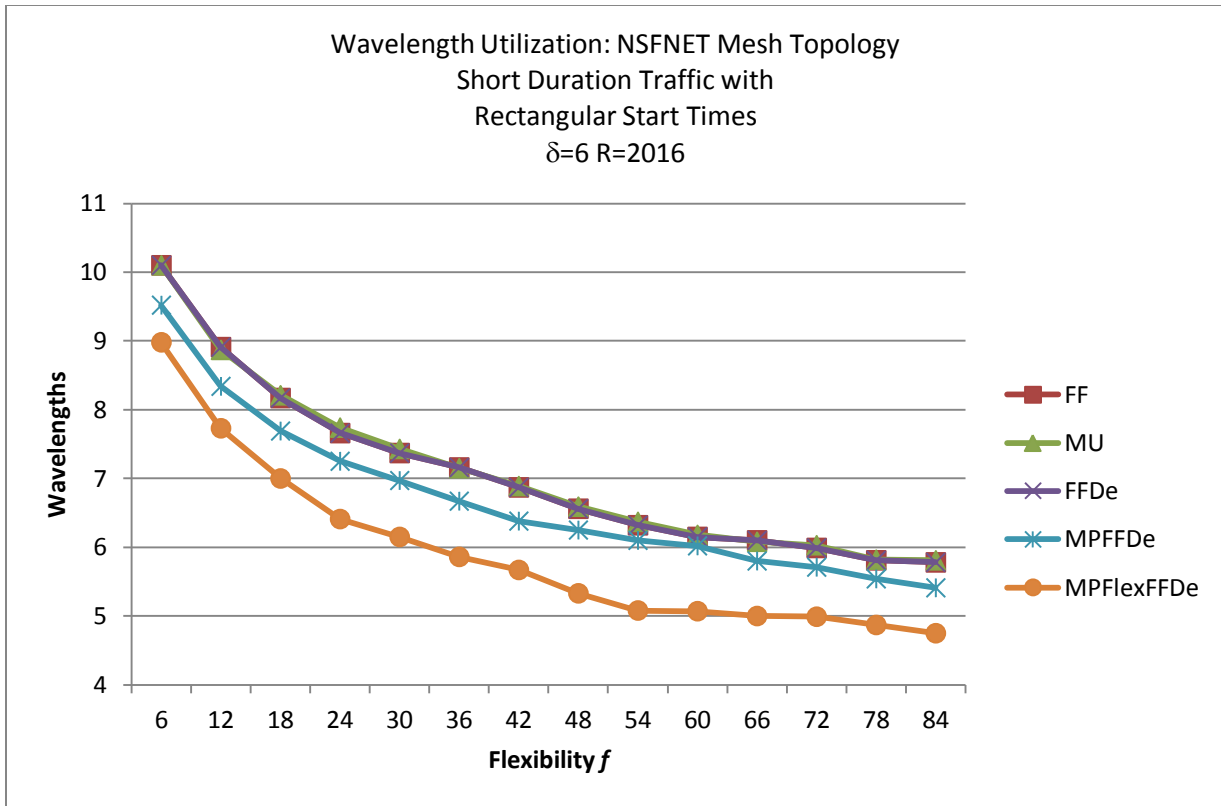


Figure 4.6. Rectangular start-time distribution using short duration traffic.

4.2.3 Modified Gaussian Traffic

Figure 4.7 again shows the average number of wavelengths used to schedule lightpath requests over the NSFNET and COST 239 mesh topologies. In this figure the traffic model now follows a modified Gaussian start-time distribution with diverse duration (i.e. $R = 504$, $\delta = 24$) traffic. Figure 4.8 is also a Gaussian distribution but the traffic duration is short (i.e. $R = 2016$, $\delta = 6$) rather than diverse.

The results show the modified Gaussian traffic distribution produces essentially the same results as the rectangular model, however, the rectangular model performs slightly better in terms of wavelength utilization.

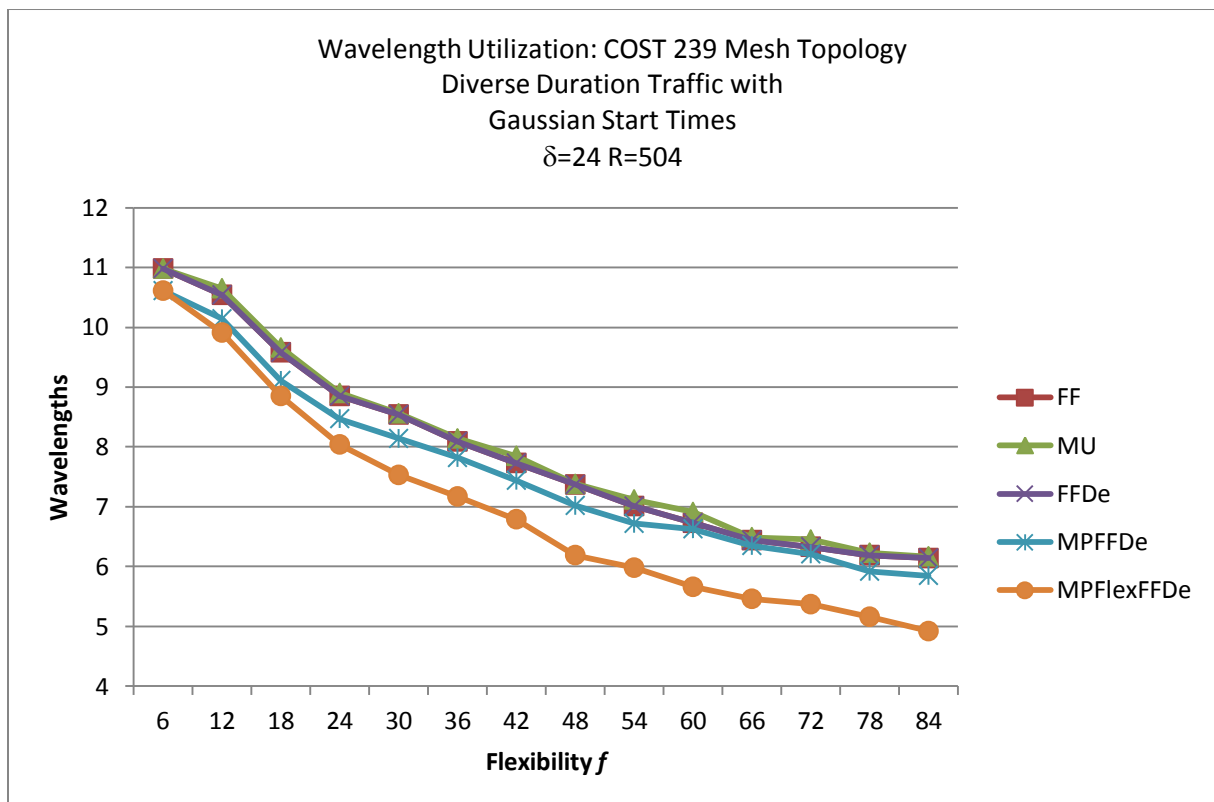
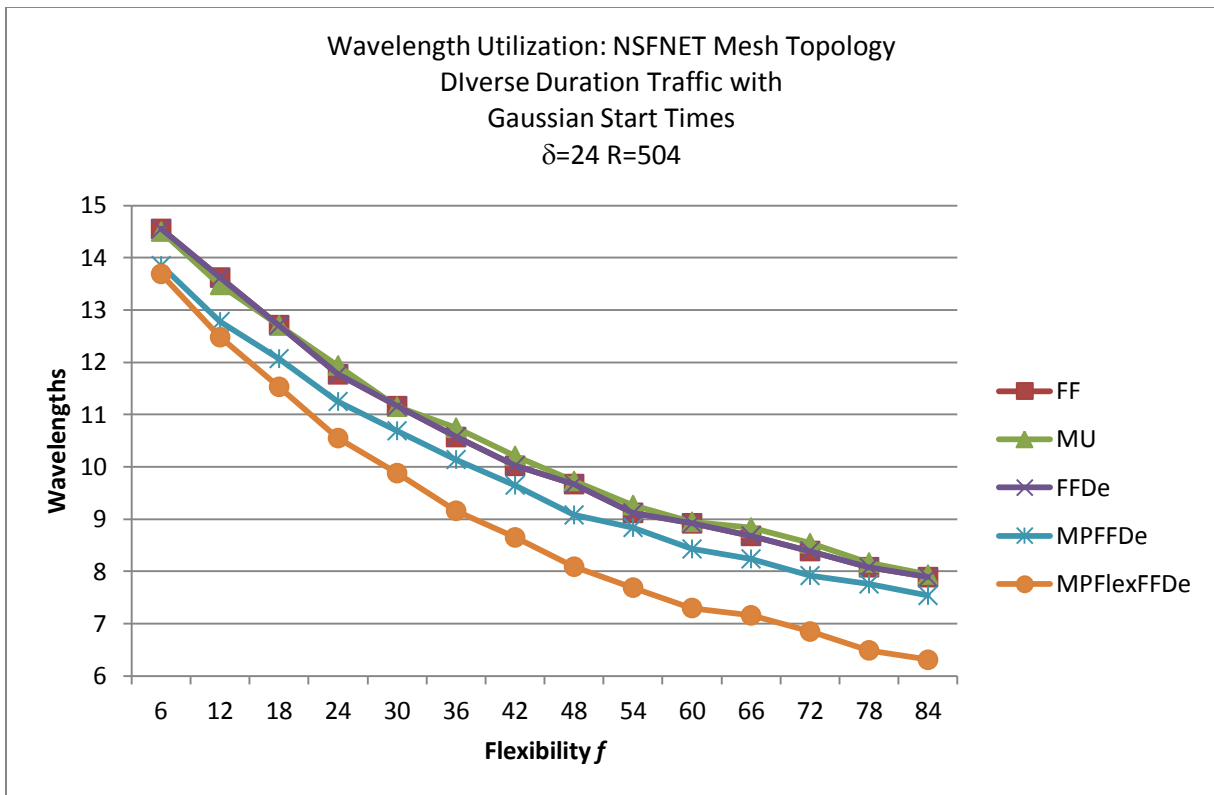


Figure 4.7. Modified Gaussian start-time distribution using diverse duration traffic.

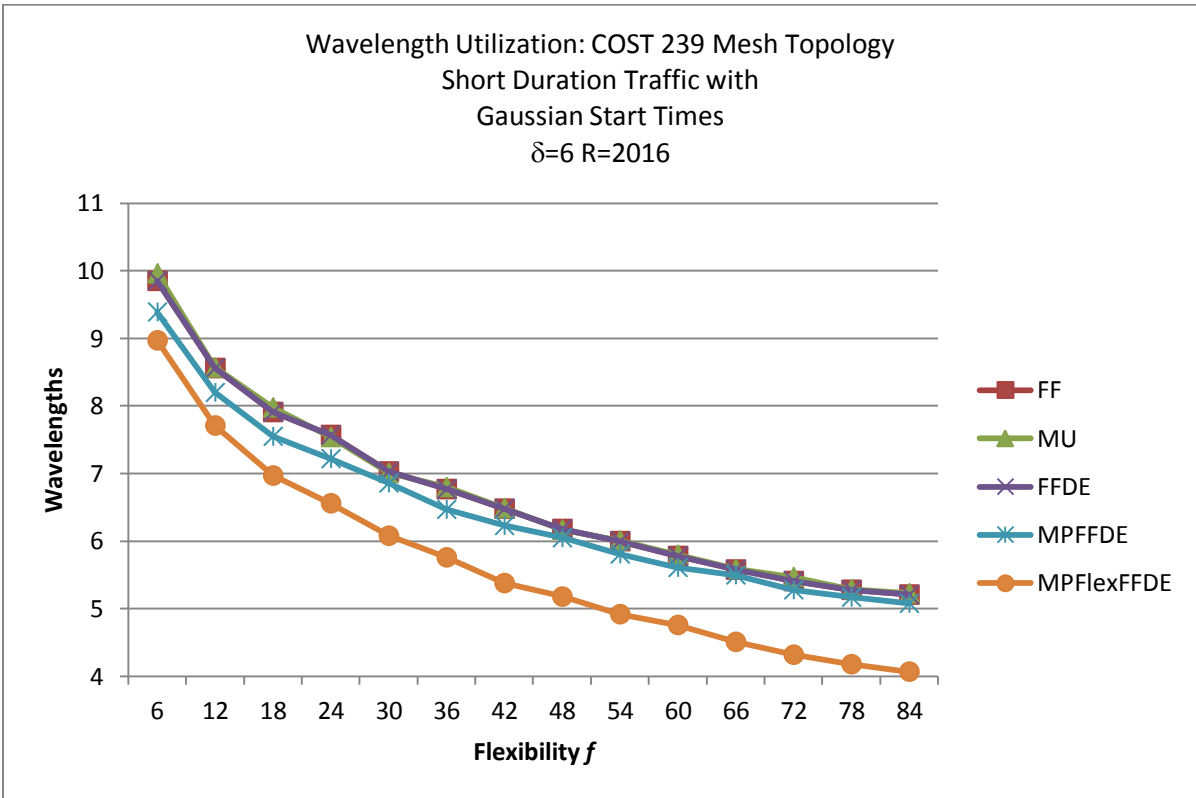
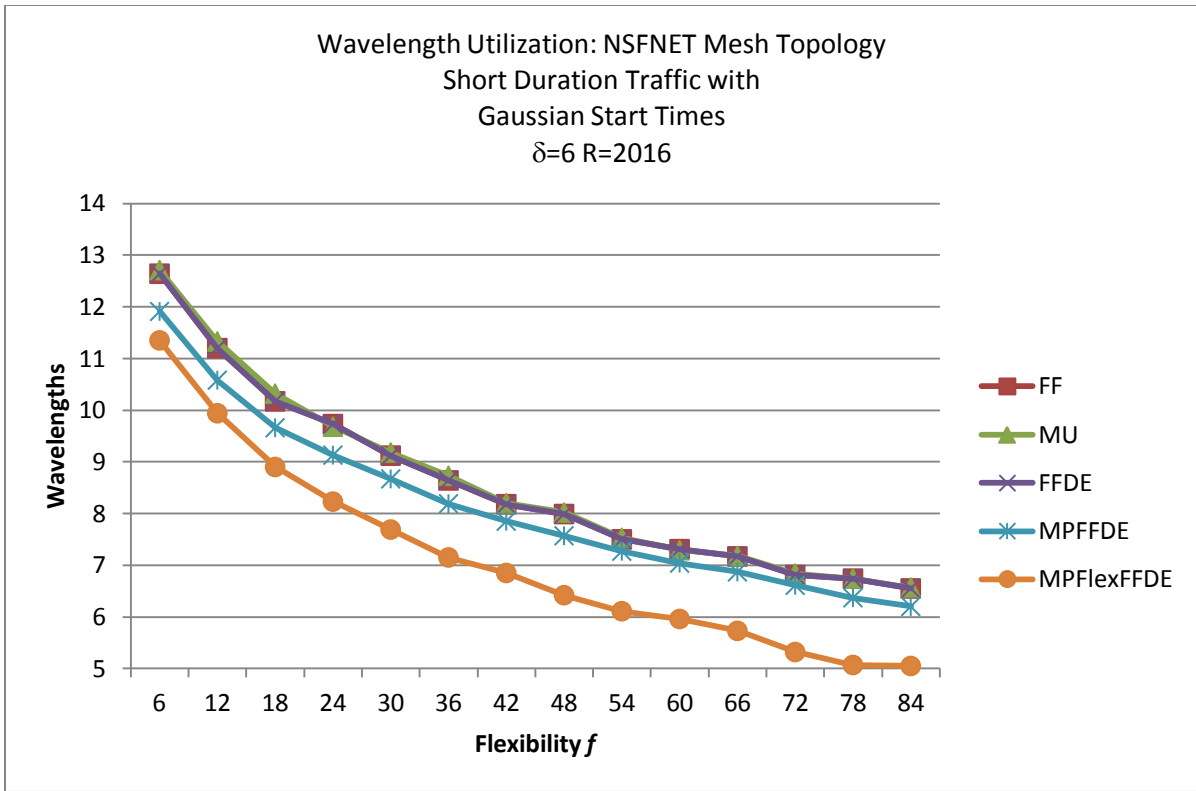
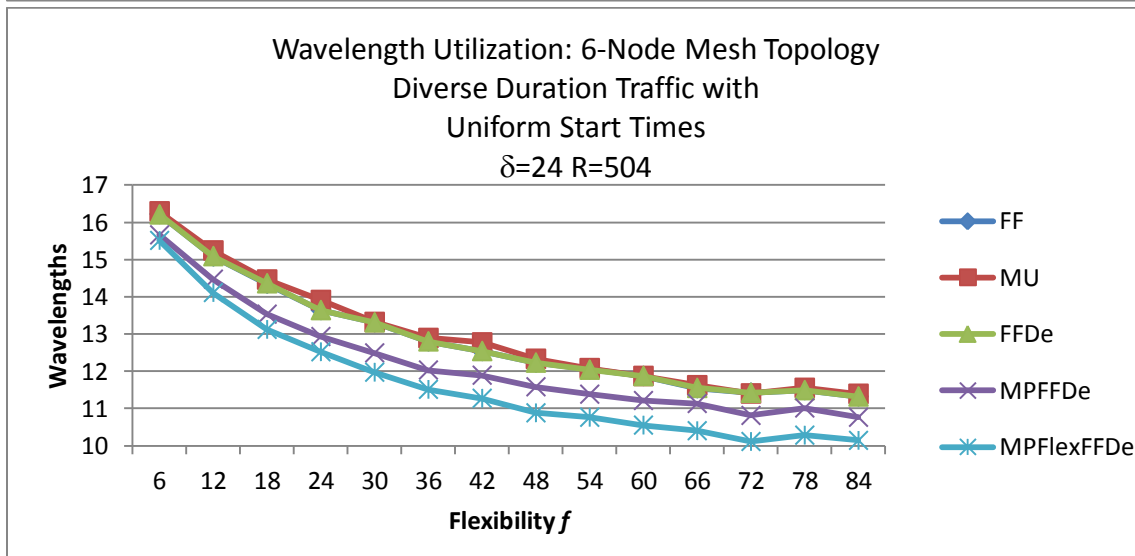
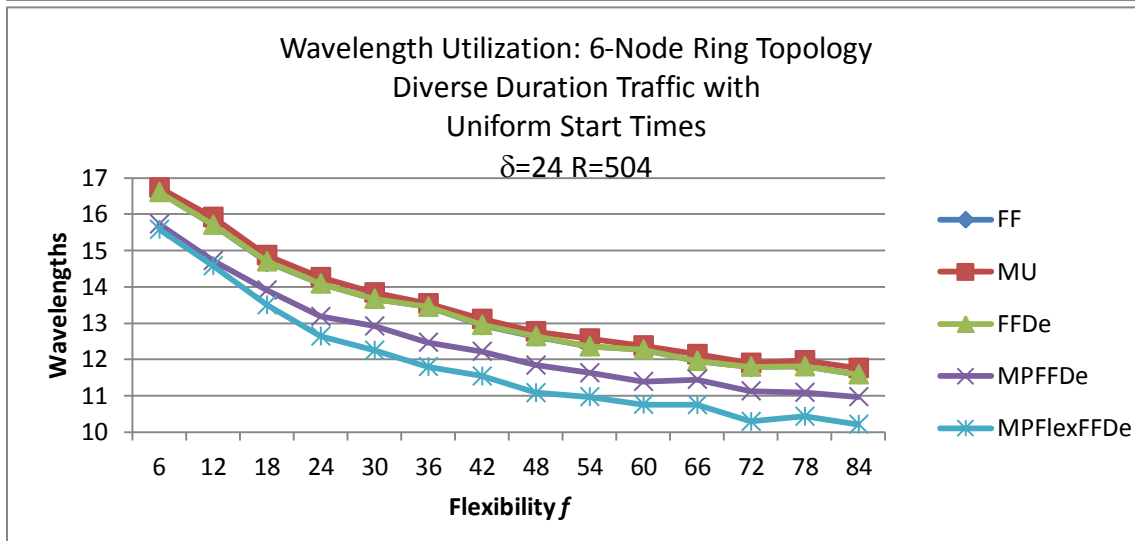
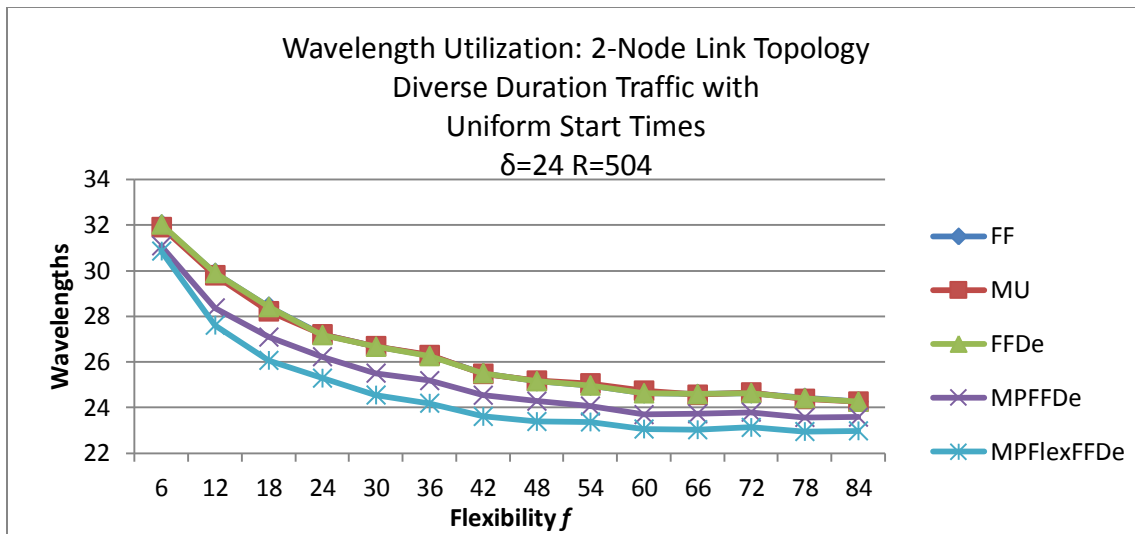


Figure 4.8. Modified Gaussian start-time distribution using short duration traffic.

4.2.4 Topology Comparison

We ran simulations on three topology classes, namely a point-to-point link, a ring, and a mesh topology. Figure 4.9 gives a comparison of the results from the simulations run over the following five topologies: a single link, a 6-node ring, a 6-node mesh (from Figure 1.2), the COST 239 mesh, and the NSFNET mesh. From the figure we see that no matter what the topology is the MPFlexFFDe heuristic is the most efficient, followed by MPFFDe, and finally FF, FFDe, and MU. This suggests there is no direct relationship between algorithm performance and topology type.



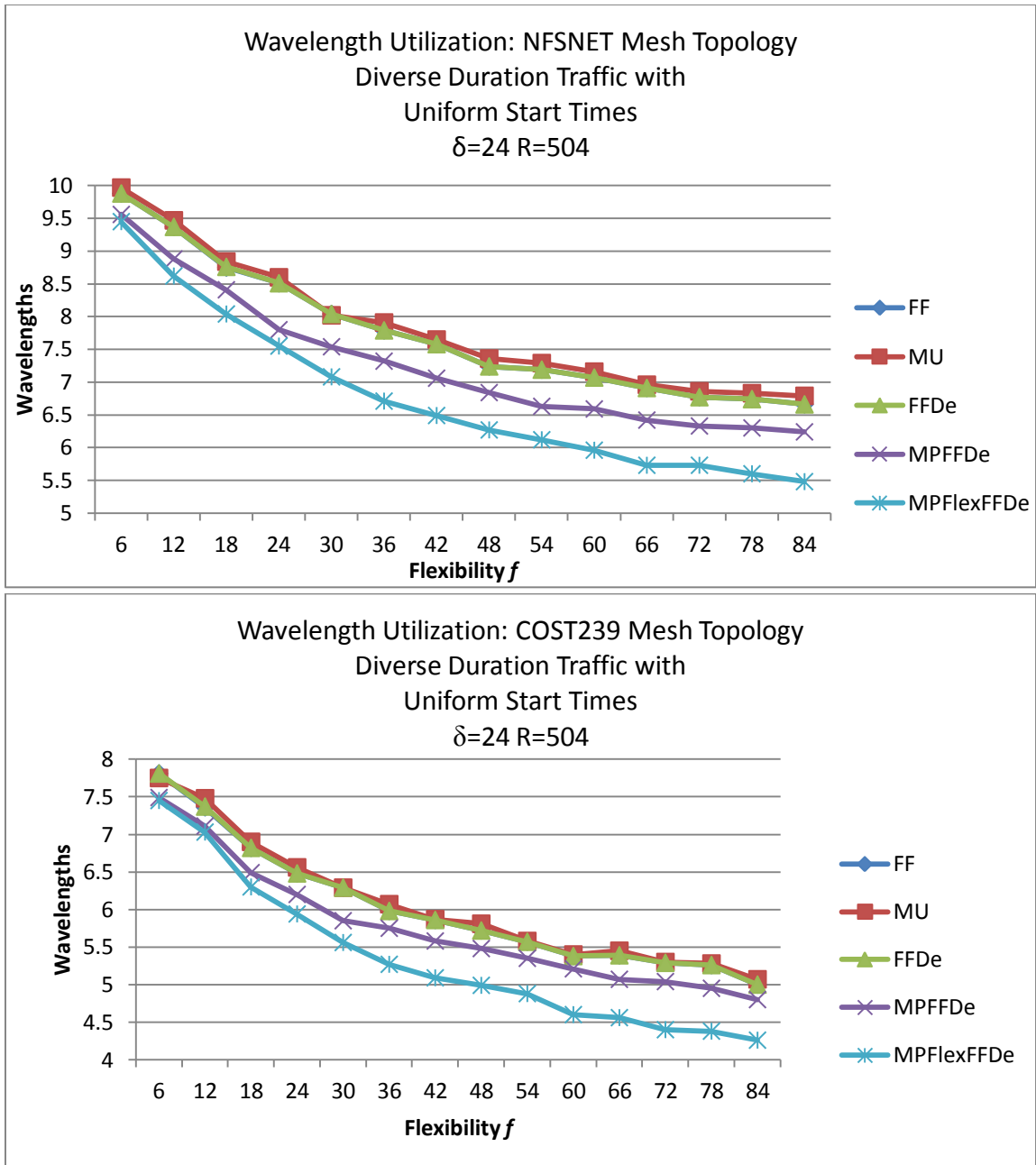



Figure 4.9. Five topology comparisons for uniform distribution.

4.3 Simulation Conclusion

In terms of algorithm performance, Table 4.1 summarizes our simulation results.

Table 4.1. Performance summary of traffic with variable flexibility.

Rank	Short Duration Traffic	Diverse Duration Traffic
Best	MPFlexFFDe	MPFlexFFDe
	MPFFDe	MPFFDe
	Similar Performance	FF, FFDe
		MU
Worst		

The lightpath request traffic generated for testing was made more realistic by randomly varying the flexibilities for each request in a given test batch. When using the more realistic traffic model, MPFlexFFDe does consistently better at scheduling lightpath requests than any other algorithm. This is true over three different traffic distributions and three topology classes.

Topology formation did not seem to make a difference in algorithm performance. The performance rankings shown in Table 4.1 still hold true regardless if the topology is a simple link, ring, or general mesh formation.

CHAPTER 5

CONCLUSION

5.1 Summary

In this thesis, we looked at the problem of scheduling periodic lightpaths with flexibility over various optical WDM network topologies. We proposed a more realistic traffic scenario in which every lightpath request could contain a different *time flexibility* value. We also proposed a new scheduling algorithm MPFlex which takes into account a request's time flexibility and determines how difficult it will be to schedule compared to other requests. Based on the scheduling difficulty, it then schedules the lightpaths in a given set starting with the most difficult request. Along with four of the best scheduling algorithms from [3] [4] simulations were completed to evaluate performance. It turns out the MPFlexFFDe algorithm produced the best results overall for every topology and traffic model. In terms of a relationship between topology type and algorithm performance, our results showed each of the five algorithms performed the same (compared to each other) regardless of topology type.

The focus of this thesis has been on offline algorithms. But some conclusions can be made about online algorithms for dynamic traffic. Consider the simulation results in Chapter 4. The simulations for the FF, MU, and FFDe algorithms assume that lightpath requests are ordered in some arbitrary way, and then scheduled one at a time in the order. This models the dynamic traffic scenario where randomly generated lightpath requests arrive sequentially. Then algorithms, such as FF, MU and FFDe, schedule lightpath requests as they arrive, and therefore operate as online algorithms. Also, once a lightpath has been schedule, it doesn't change.

The simulations for the MPFFDe and MPFlexFFDe algorithms correspond to another scenario. This scenario also models the dynamic traffic case where randomly generated lightpath requests arrive sequentially. However, lightpaths that are already scheduled can be rescheduled. The difference in performance between the FF, MU, and FFDe algorithms and the MPFFDe and MPFlexFFDe algorithms indicate the improvement by allowing lightpaths to be rescheduled. The simulations in Chapter 4 show that, in the case of dynamic traffic, allowing lightpaths to be rescheduled can lead to significant reductions in the number of required wavelengths.

5.2 Future Work

Although we considered several aspects of the scheduling problem in this thesis, more work still needs to be done. Statistical and empirical analysis, such as the work done in [2] needs to be completed for traffic models with varying flexibilities. Additionally, producing a formula that accurately describes the relationship between the topology type, time flexibility, and wavelength usage should be developed.

REFERENCES

- [1] H. Zang, J. P. Jue, B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks," *Opt. Net.Mag.*, vol. 1, no. 1, pp. 47–60, Jan. 2000.
- [2] W. Su, G. Sasaki, "Scheduling of periodic transfers with flexibility," *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 1-3, 2003.
- [3] L. Ye, "Online scheduling of periodic lightpath request with flexibility," *UH Manoa Electrical Engineering Department Thesis Archive*, Aug. 2007.
- [4] D. Teng, "Scheduling algorithms in a ring wdm network," *UH Manoa Electrical Engineering Department Thesis Archive*, Dec. 2009.
- [5] X. Yang, L. Shen, A. Todimala, B. Ramamurthy, T. Lehman, "An efficient scheduling scheme for on-demand lightpath reservations in reconfigurable wdm optical networks," *Optical Fiber Communication Conference, 2006 and the 2006 National Fiber Optic Engineers Conference. OFC 2006*, vol. 3, Mar. 5-10, 2006.
- [6] L. Shen, X. Yang, A. Todimala, B. Ramamurthy, "A two-phase approach for dynamic lightpath scheduling in wdm optical networks," *Communications, 2007. ICC '07. IEEE International Conference*, Jun. 24-28, 2007
- [7] L. Shen, A. Todimala, B. Ramamurthy, X. Yang, "Dynamic lightpath scheduling in next-generation wdm optical networks," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, Apr. 23-29, 2006
- [8] J. Hecht , "Ultrafast fibre optics set new speed record,"

URL: <http://www.newscientist.com/article/mg21028095.500-ultrafast-fibre-optics-set-new-speed-record.html>

- [9] L. Ruan, F. Tang, "Survivable ip network realization in ip-over-wdm networks under overlay model," *Computer Communications, Elsevier B.V.*, vol. 29, no. 10, pp. 1772-1779, 2006.
- [10] S. Pasqualini, A. Iselt, A. Kirstädter, A. Frot, "Mpls protection switching vs. ospf rerouting: a simulative comparison,"
URL: http://www.lkn.ei.tum.de/~akirstaedter/papers/2004_QofIS_MPLS_OSPF.pdf
- [11] The Cooperative Association for Internet Data Analysis, "Caida san jose passive network monitor,"
URL: <http://www.caida.org/data/realtime/passive/>
- [12] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, R. Douville, "Routing and wavelength assignment of scheduled lightpath demands," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 8, pp. 1232-1240, 2003.
- [13] B. Wang, T. Li, X. Lou, Y. Fan, "Traffic grooming under a sliding scheduled traffic model in wdm networks," *Proc. IEEE Workshop on Traffic Grooming in WDM Networks*, San Jose, Oct. 29, 2004.
- [14] B. Wang, A. Deshmukh, "An all hops optimal algorithm for dynamic routing of sliding scheduled traffic demands," *IEEE Communications Letters*, vol. 9, no. 10, Oct. 2005.
- [15] J. Zheng, H. T. Mouftah, "Routing and wavelength assignment for advance reservation in wavelength-routed wdm optical networks," *Communications, 2002. ICC 2002. IEEE International Conference on* , vol. 5, pp. 2722- 2726, 2002

- [16] R. A. Guerin, A. Orda, "Networks with advance reservations: the routing perspective," *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* , vol. 1, no. 1, pp.118-127, 2000