

Batch Wavelength Assignment in All-Optical Central-Switch Networks

by

Won Sang Yoon

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1997

© Massachusetts Institute of Technology 1997. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
August 27, 1997

Certified by
Dr. Richard A. Barry
Staff Member, Advanced Networks Group, MIT Lincoln Laboratory
Thesis Supervisor

Certified by
Robert G. Gallager
Fujitsu Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Batch Wavelength Assignment in All-Optical Central-Switch Networks

by

Won Sang Yoon

Submitted to the Department of Electrical Engineering and Computer Science
on August 27, 1997, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

We present a study of dynamically assigning wavelengths to *batches* of call requests in an all-optical wavelength-division multiplexed, central-switch network. All known previous studies of dynamic wavelength assignment have analyzed *sequential* methods, where calls are established one at a time. This study of batch assignment is primarily motivated by the scenario of dynamically establishing and removing groups of calls forming *logical topologies* of higher-level networks on top of an all-optical backbone network. Another motivation is the possibility of obtaining a more efficient wavelength assignment by deliberately *waiting and collecting* calls before establishing them as a group.

We characterize the call requests by the maximum number of connections through any link, designated as L , and for batch assignment we assume a *minimum batch size* of B calls. Our objective is to find the minimum number of wavelengths needed to guarantee that any sequence of requests can be established. We first obtain the optimal solution by assuming *a priori* knowledge of all requests. We then prove that sequential assignment almost always requires twice this optimal number of wavelengths. Finally, we propose a class of batch assignment algorithms and show that for small networks and large L , they can do no better than sequential methods unless B is very large. For larger networks and small L , however, our analysis shows that batch assignment potentially requires fewer wavelengths than sequential methods for smaller values of B .

Thesis Supervisor: Dr. Richard A. Barry

Title: Staff Member, Advanced Networks Group, MIT Lincoln Laboratory

Thesis Supervisor: Robert G. Gallager

Title: Fujitsu Professor of Electrical Engineering and Computer Science

Acknowledgments

I would first of all like to thank my advisor, Dr. Richard Barry, for his patience and meticulous attention to detail. His timely advice and numerous proofreads nurtured and refined both my thesis and my approach to research. I thank Prof. Robert Gallager for his insight and support throughout my research and Dr. Steven Finn for his comments on the results of my thesis. I am grateful to Group 65 of Lincoln Laboratories for providing me with the opportunity to work in a stimulating and supportive atmosphere. Finally, I would like to thank my parents; words cannot adequately express my appreciation for all they have given to me.

Contents

1	Introduction	8
1.1	Background	8
1.2	A Motivating Example	10
1.3	Overview of Previous Research	12
1.4	Outline of Thesis	16
2	The Problem Formulation	17
2.1	The Network	18
2.2	The Traffic	19
2.3	The Analysis	19
3	Static Wavelength Assignment	22
4	Sequential Wavelength Assignment	27
4.1	Strict-Sense Nonblocking	28
4.2	Greedy Sequential Algorithms	31
4.2.1	Fully-Dynamic Traffic	32
4.2.2	Semi-Dynamic Traffic	36
4.3	A Simple Non-Greedy Algorithm	39
4.4	Summary of Sequential WA	45
5	Batch Wavelength Assignment	47
5.1	Defining the Problem	47
5.2	Strict-Sense Nonblocking	49

5.2.1	Fully-Dynamic Traffic	50
5.2.2	Semi-Dynamic Traffic	55
5.2.3	Batch-Dynamic Traffic	56
5.2.4	Summary of SSNB	57
5.3	Greedy Batch Algorithms	58
5.3.1	Fully-Dynamic Traffic	59
5.3.2	Semi-Dynamic Traffic	67
5.3.3	Batch-Dynamic Traffic	73
5.4	Summary of Batch WA	77
6	Conclusions and Further Research	80
A	Alternate Proof of Static WA	83
B	Routing in Clos Networks	85

List of Figures

1-1	The connection matrix	11
1-2	Sequential vs. batch assignment	12
1-3	Sequential vs. batch assignment (continued)	13
1-4	One possible organization of wavelength assignment	14
2-1	A centralized-switch WDM network	18
3-1	Optimal static WA algorithm	25
4-1	Example requiring $2L - 1$ wavelengths	30
4-2	Example requiring lower bound for fully-dynamic greedy sequential WA	34
4-3	Upper and lower bounds on W_{NB} for fully-dynamic greedy sequential WA	35
4-4	Example requiring lower bound for semi-dynamic greedy sequential WA	38
4-5	Lower and upper bounds for semi-dynamic and fully-dynamic greedy sequential WA	39
4-6	Plot of bounds for greedy sequential WA	40
4-7	Pre-assigned lookup table	41
4-8	Worst-case for greedy sequential WA with no repeated calls	43
4-9	“Lookup-table” vs. greedy WA for no repeated calls	44
5-1	Plot of $W_{SSNB}^{batch/fd}(B)$ as a function of B	51
5-2	Chains of reassignment in batch WA	53
5-3	Example of greedy sequential vs. greedy batch WA	59
5-4	Lower bound on $W_{NB-greedy}^{batch/fd}(B)$	61

5-5	Example for the lower bound on fully-dynamic greedy batch	62
5-6	Upper and lower bounds on $W_{NB-greedy}^{batch/fd}(B)$	66
5-7	Lower bound on $W_{NB-greedy}^{batch/sd}(B)$	68
5-8	Example proving lower bound on semi-dynamic greedy batch	69
5-9	Lower bound vs. upper bound for semi-dynamic greedy batch	72
5-10	Lower bound on $W_{NB-greedy}^{batch/bd}(B)$	74
5-11	Example proving lower bound on batch-dynamic greedy batch	75
5-12	Comparison of lower bounds and upper bounds for greedy batch WA	79

Chapter 1

Introduction

As all-optical networks become practical, there has been extensive research aimed at understanding how to manage and control these networks to best utilize their resources. In particular, the issue of *routing and wavelength assignment* in all-optical networks has been a rich bed of interesting studies. This is largely due to the fact that the constraints imposed by the all-optical nature of these networks add new difficulties which are not present in the problems of routing and channel assignment for classic circuit-switched networks. The canonical problem can be stated as: given a set of connection requests to establish in the network, we wish to assign both a *route* and a *wavelength* to each connection such that some criterion is optimized, while satisfying all necessary constraints.

Before describing this problem in more detail, we first present a brief introduction to all-optical networks and describe the physical constraints which define the framework of this study. We then give an overview of previous research in this area and proceed to formulate the specific problem which we have studied.

1.1 Background

All-optical networks are characterized by the property that connections between end-users (or network access stations) are established entirely by optical signals throughout the network, with no intermediate optical-to-electronic signal conversion. At this

time, the most prevalent transmission method in such all-optical networks is based on wavelength-division multiplexing (WDM), where a set of orthogonal wavelengths provides multiple simultaneous channels on every strand of fiber. The traffic-carrying capacity of these WDM networks is directly related to the number of unused wavelengths available on each fiber link, which in turn is determined by the particular assignment of wavelengths to existing connections. Thus, the choice of wavelength assignment method can have a significant impact on the capacity of the overall network. The need for a careful choice of a wavelength assignment method is underscored by the fact that current commercial systems are currently limited to at most a few dozen wavelengths per fiber¹.

This goal of efficiently assigning wavelengths in an all-optical network is limited by two constraints, as defined by Stern in 1991 [22]:

1. **Wavelength Inseparability:** this constraint requires that any two calls whose paths share a common fiber in the network must be assigned different wavelengths. This is a result of the fact that two signals which are transmitted on the same wavelength over the same fiber are physically indistinguishable to wavelength-selective switches and receivers.
2. **Wavelength Continuity:** this requires that the same wavelength be used by a signal on every link of its path through the network. The presence of wavelength converters at intermediate nodes can relieve this constraint, but at the expense of increased management complexity and device cost. If we consider the case of wavelength converters at every node in the network, then the wavelength-assignment problem would be completely equivalent to channel assignment in a pure circuit-switched network. In our study, we will assume the worst-case wavelength assignment scenario of no wavelength conversion.

Given a call request in the network, these two constraints restrict our set of assignable wavelengths to those which are unused on every link of the call's path. If

¹although the bandwidth of an optical fiber potentially allows many more.

no such wavelength is available (assuming there are a limited number of wavelengths), then the request is blocked. One of the primary goals of wavelength assignment algorithms is to minimize the probability of blocking future requests by judiciously assigning wavelengths to current ones.

In general, given a set of connection requests to establish in a WDM network, we must assign both a wavelength and a route to each call. In our study, we will decouple the two problems and focus on trying to fully understand the problem of *wavelength assignment* for calls with fixed routes.

In order to illustrate the basic principles of this problem, we present a simple example which will also serve to motivate the purpose of our study.

1.2 A Motivating Example

Figure 1-1 shows a centralized-switch network with 5 access stations and 2 unidirectional fibers per link. The switch is wavelength-selective and can route a call on a given wavelength from any input port to any output port on the same wavelength. Assume that each access station has one receiver and one transmitter for each wavelength. For convenience, we can represent this network and its connections by a *connection matrix*, shown alongside in Figure 1-1. Those symbols in an entry (i, j) represent the wavelengths assigned to the corresponding connections from station i to j .

Now consider a sequence of randomly arriving connection requests, shown in Figure 1-2-a, and assume we assign wavelengths to them one-by-one as they arrive. Suppose our algorithm is to just assign the first available wavelength to a call (assuming the wavelengths are in some order), referred to as the *first-fit* method. Due to the wavelength-inseparability constraint, we cannot assign a wavelength which already appears in either the row or the column of the call request. Under this sequential assignment rule, we see that the sixth request, from station 4 to 2, must be assigned a third wavelength, green. Although there are at most two calls in any column or

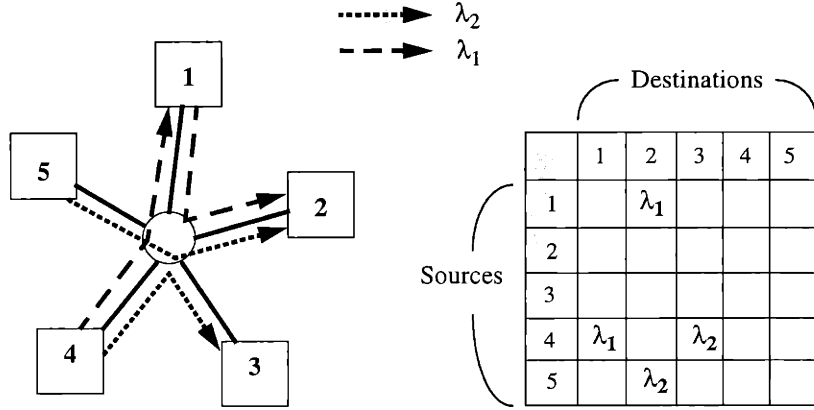


Figure 1-1: The connection matrix

row in the matrix, we need three wavelengths to sequentially establish this particular sequence of calls.

Now assume that we assign wavelengths to all six requests together, rather than sequentially. This may be a result of the requests arriving naturally as a group, or we may wish to deliberately *wait and collect* the independent call arrivals before assigning wavelengths to them. In any case, if the six requests are known initially, then Figure 1-2-b shows two possible assignments which require only two wavelengths. So we see that dynamically assigning wavelengths to *groups* of calls, rather than one at a time, could allow us to assign wavelengths more efficiently. Now if we compare the two batch assignments of Figure 1-2-b, there does not seem to be an obvious preference. Continue this example and assume that another batch of four calls arrive (or assume we wait and collect the next four calls), as in Figure 1-3. The sequential assignment and the first batch assignment both need a third wavelength, green. The second batch assignment, however, can establish all four calls still using only two wavelengths. This shows how a particular choice of wavelengths for a group of calls can affect the number of wavelengths needed in the future.

There are other performance criteria besides minimizing the number of wavelengths required to establish all requests. Another possibility is to allow some blocking (typically by assuming a finite number of wavelengths) and try to minimize the

Sequence of calls:
1-2, 2-3, 5-1, 4-1, 3-5, 4-2

(a) F-F Sequential WA:

1-2: red
2-3: red
5-1: red
4-1: blue
3-5: red
4-2: green

	1	2	3	4	5
1		R			
2			R		
3					R
4	B	G			
5	R				

(b) Batch WA # 1:

1-2: red
2-3: blue
5-1: blue
4-1: red
3-5: red
4-2: blue

	1	2	3	4	5
1		R			
2			B		
3					R
4	R	B			
5	B				

Batch WA # 2:

1-2: red
2-3: red
5-1: blue
4-1: red
3-5: red
4-2: blue

	1	2	3	4	5
1		R			
2			R		
3					R
4	R	B			
5	B				

Figure 1-2: Sequential vs. batch assignment

number of calls blocked. In our example, if we assumed only two wavelengths were available (red and blue), then the calls which were assigned green would instead have to be blocked.

These examples have shown us the possible benefits, and complications, of assigning wavelengths to calls in batches. Before delving into this topic any deeper, we first take a step back and present a brief summary of other research areas in wavelength assignment and where our study fits in.

1.3 Overview of Previous Research

There have been numerous studies of wavelength assignment (WA) in WDM networks, for various network topologies, traffic characterizations, and performance metrics. One possible organization of these works is shown in Figure 1-4. We will briefly

Next sequence of calls: 1-3, 2-5, 3-4, 5-4

(a) **F-F Sequential WA:**

1-3: blue
2-5: blue
3-4: blue
5-4: green

	1	2	3	4	5
1		R	B		
2			R		B
3				B	R
4	B	G			
5	R			G	

(b) **Batch WA # 1:**

1-3: green
2-5: green
3-4: blue
5-4: red

	1	2	3	4	5
1		R	G		
2			B		G
3				B	R
4	R	B			
5	B			R	

Batch WA # 2:

1-3: blue
2-5: blue
3-4: blue
5-4: red

	1	2	3	4	5
1		R	B		
2			R		B
3				B	R
4	R	B			
5	B			R	

Figure 1-3: Sequential vs. batch assignment (continued)

describe the different scenarios that have been studied and list the major results.

- **Static WA:** assume that all requests are known *a priori* and the traffic never changes.

1. *Nonblocking:* guarantees that every call is established, and the goal is to minimize the number of wavelengths needed to satisfy all requests. This problem of wavelength assignment with the objective of minimizing the number of wavelengths used was proven by Chlamtac in 1992 to be equivalent to that of graph-coloring and therefore NP-complete for an arbitrary network topology [10]. Various heuristics for static nonblocking WA were presented by Chlamtac [10], Wauters [23], and Banerjee [4], with simulation results. Baroni [5] and Wischik [24] analyzed the number of required wavelengths as a function of network topology. Ring and tree networks were

studied by Ramaswami [19] and Gerstel [13], and they obtained bounds on the number of wavelengths required as a function of the maximum number of calls on a link. Aggarwal [2] obtained bounds on the number of wavelengths needed as a function of “congestion” (maximum calls on a link) and “dilation” (maximum path length) of the network, but did not focus on any specific network topologies.

2. *Blocking*: in this scenario, call requests may be blocked, and the objective is usually to minimize the proportion of blocked calls. Chlamtac [10] presented simulation results of the dependence of blocking probability on the number of available wavelengths. Ramaswami [20] obtained analytical bounds on the maximum number of calls allowed per wavelength, and both Ramaswami [20] and Mokhtar [17] formulated a linear program to maximize the amount of traffic subject to the number of wavelengths available.

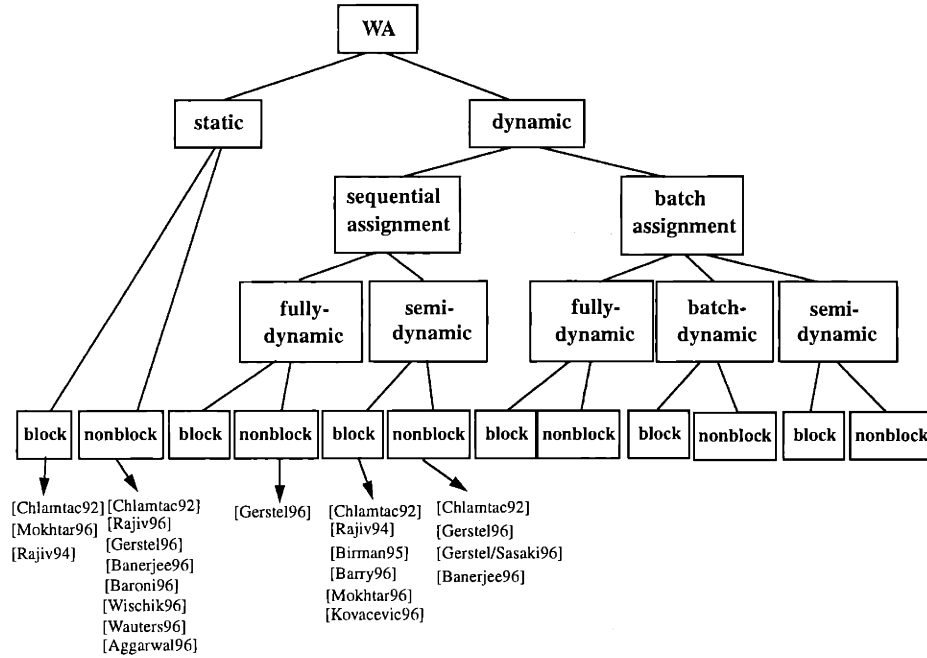


Figure 1-4: One possible organization of wavelength assignment

- **Dynamic WA**: the alternative to static WA is to assume that calls may arrive and depart to and from the network randomly. All previous studies have only

considered *sequential* wavelength assignment, where calls are established one by one without knowledge of the next request. The other possibility, and the focus of our work, is to assign wavelengths to *batches* of dynamically arriving and departing calls. We can further classify studies based on the assumptions of call departures:

1. *Fully-dynamic*: assume that calls can individually disconnect from the network at random. Some works assumed a *blocking* scenario, and tried to minimize the blocking probability by using various heuristics. Chlamtac [10], Birman [1], and Acampora [16] have studied *First-Fit* assignment, where the first available wavelength is assigned to the next call, assuming that the wavelengths are in some order. Acampora [16] and Mokhtar [17] studied *Random* assignment, where the next wavelength assignment is chosen randomly from the pool of available wavelengths. Barry and Subramaniam [6] proposed the *Max-Sum* algorithm, which chooses that wavelength assignment which minimizes a cost function in the *subsequent* network state. Other studies assumed a *non-blocking* network and tried to minimize the number of wavelengths required, as done by Banerjee [4], Chlamtac [10], Gerstel [13], and Sasaki [14]. The latter two obtained lower bounds on the number of wavelengths needed for ring and tree networks.
2. *Semi-dynamic*: assume that once calls are established in the network, they never disconnect. Very little work has been done in this context. Gerstel [13] addressed this case for a ring network, presenting bounds on the maximum number of wavelengths needed as a function of the traffic load (maximum number of calls on any link). No other studies of the semi-dynamic case are known.
3. *Batch-dynamic*: another possibility, only applicable for batch wavelength assignment, is for calls to randomly leave the network only in the same batches in which they were established. We do not know of any previous studies which have addressed this possibility.

In summary, static WA is usually constrained by the algorithmic complexity of finding a set of wavelength assignments for a possibly large number of calls at once. Sequential WA, on the other extreme, is limited in performance by the uncertainty of future requests and the difficulty of “planning ahead”. Our study of dynamic batch assignment seems to lie somewhere between these two cases. Aggregating call arrivals into “manageable”-sized groups and assigning wavelengths to each group at a time may give us the flexibility to find a more efficient assignment than the sequential case, but without the complexity of having to deal with the entire set of requests at once, as in the static case. The less optimistic view is that grouping call arrivals will increase the complexity over sequential WA, while yielding a less efficient assignment than static WA (which has knowledge of the entire set of requests). We will examine the relative benefits and disadvantages of batch WA in terms of these tradeoffs.

1.4 Outline of Thesis

In the next chapter, we will define the framework of the problem which we study, and describe our analytical approach. Then, we begin our analysis with *static WA* in Chapter 3, which will give us the optimal solution to our problem. In Chapter 4, we will study the case of *sequential WA*, where calls are assigned wavelengths one at a time. We then propose the case of *batch WA* in Chapter 5 and analyze how its performance compares with that of sequential WA and static WA. Finally, we present concluding remarks and some possible further directions of research in Chapter 6.

Chapter 2

The Problem Formulation

Dynamic wavelength assignment has previously been studied for the case of establishing calls one at a time, and several algorithms have been proposed [10, 6, 13, 1]. It is not well understood, however, how wavelengths should be assigned for batch arrivals and how the performance of batch assignment compares with sequential methods. There are several motivations for studying the batch assignment case. First of all, if an all-optical network is used as a backbone for higher-layer networks, then the call requests into the optical network will most likely arrive as a *set of connections* representing some higher-layer network's logical topology. In time, the higher-layer networks may change and new logical topologies may be added in the optical backbone or existing ones can disconnect and possibly re-connect in a different configuration, as described by Bala, *et al* in [3]. Thus, it seems natural in this environment to study the possibility of assigning wavelengths to batches of calls which may dynamically arrive and depart. Even if this were not the case, however, we have already seen that it could be advantageous for a network manager to deliberately wait and collect individually arriving calls before assigning wavelengths to them as a group.

In studying the problem of batch wavelength assignment, there are various possibilities of network topology, traffic statistics, and performance metrics. We describe our assumptions next.

2.1 The Network

We will study a network topology of a central, wavelength-selective switch connecting N network access stations, as shown in Figure 2-1. These network access stations may be end-users or other central switches in a hierarchical network structure. The central switch can route a call from any input port on any wavelength to any output port on the same wavelength. We assume no wavelength conversion in the central switch. We have chosen to study a central-switch network since its fixed routing structure allows us to focus on understanding the problem of wavelength assignment. Practically, this network could serve as a central switch for local exchange networks and might be one of the first AONs implemented commercially. We will assume that each link between the central switch and network access stations has one uni-directional fiber in each direction, and that each station has one transmitter and one receiver for each wavelength (as shown in Figure 2-1). We assume that only point-to-point connections are allowed, and that each user can transmit and receive as many simultaneous calls as allowed by the wavelength constraints.

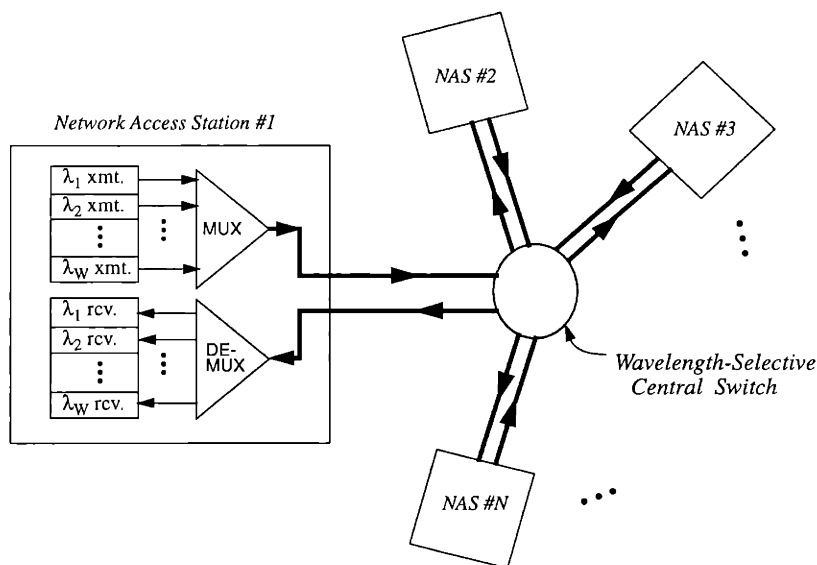


Figure 2-1: A centralized-switch WDM network

2.2 The Traffic

We assume that calls may be established in the network all at once (*static WA*), one-by-one (*sequential WA*), or in groups at a time (*batch WA*). In both sequential and batch WA, we will assume that calls may be *fully-dynamic* (random disconnections) or *semi-dynamic* (no disconnections). In batch WA, we will also consider the third traffic characteristic of *batch-dynamic*, where calls may only disconnect in the same batches in which they were established. We do not assume any particular arrival or departure statistics, but instead characterize the call requests only in terms of the *maximum number of calls sharing any link in the network*, referred to as the “maximum link load”, or L . Thus, for any given sequence of connections and disconnections, we will assume that calls always satisfy the maximum link load condition. In terms of actual network implementation, we might view this as a result of a call-admission scheme which pre-rejects any request that would violate the maximum load on any link, regardless of the availability of a wavelength on that link (then we would not consider such a rejection to be a “blocked call” for wavelength assignment purposes). In any case, we assume that this maximum link load L is our only characterization of the call arrivals throughout this study.

2.3 The Analysis

The primary objective of our study is to find *the minimum number of wavelengths needed to guarantee no blocking*, denoted as W_{NB} . There are three types of nonblocking operation which have been defined in literature, introduced by C. Clos in 1953 and V. Benes in 1962.

- **Rearrangeably NB:** assume that any number of previous wavelength assignments may be rearranged before establishing the next request [7]. There must be enough wavelengths to guarantee no blocking using the optimal WA algorithm.
- **Wide-sense NB:** assume that calls *may not* be reassigned wavelengths once they are established in the network. The network must have enough wavelengths

to ensure that the optimal WA algorithm never blocks a call [7]. In general, this will require more wavelengths than the rearrangeable case.

- **Strict-sense NB:** no reassignments are allowed, and now there must be enough wavelengths to ensure that there will always be some valid wavelength assignment for the next request, regardless of the previous wavelength assignments [11]. This is the strongest of the three conditions and requires the most wavelengths.

We can relate these nonblocking conditions on the network with our previous characterizations of the call requests. For the case of *static WA*, the network must be rearrangeably nonblocking, since all requests are known *a priori* and any number of wavelength assignments may be changed before the entire set of calls is finally established in the network. In the dynamic cases of *sequential WA* and *batch WA*, we will study the strict-sense nonblocking condition and also study the potential benefits of a particular class of algorithms for different departure characteristics. We do not consider wide-sense nonblocking operation.

To be more precise, we define the following notation which will be used throughout this study:

$W_{NB-type}^{arr/dep}$ = the minimum number of wavelengths needed to guarantee no blocking for any sequence of calls with arrival characteristic “arr” and departure characteristic “dep” under nonblocking operation “NB-type”, where

“arr” $\in \{ static, sequen, batch \}$

“dep” $\in \{ fd, sd, bd \}$ and

“NB-type” $\in \{ SSNB, NB-alg, RNB \}$.

For example, “ $W_{SSNB}^{sequen/fd}$ ” represents the minimum number of wavelengths needed to achieve strict-sense nonblocking for *sequential* call requests with *fully-dynamic* departures. (Note that any time we omit subscripts or superscripts from this expression,

it will imply that the argument holds under any assumption).

An alternative to analyzing nonblocking operation would be to allow some blocking and compare the blocking probabilities of wavelength assignment algorithms, or to study the number of wavelengths required to achieve a specified blocking probability. This type of analysis, however, is rather restrictive since it may depend on a particular assumption of arrival statistics, and we do not consider them in this study.

Now that we have laid the framework for our study, we proceed with the analysis. In the next three chapters, we present results for static, sequential, and batch wavelength assignment in a central-switch network. We compare the bounds on the number of wavelengths required for no blocking in these three scenarios and, in some cases, describe wavelength assignment algorithms which achieve these bounds.

Chapter 3

Static Wavelength Assignment

In this chapter, we assume *a priori* knowledge of the entire set of call requests, and assume the traffic never changes thereafter. The problem of assigning wavelengths to these calls is then purely combinatorial, and we seek the minimum number of wavelengths needed to establish the given set of requests. In this scenario, the network must be *rearrangeably* nonblocking, so that the minimum number of wavelengths needed for no blocking in static WA is the same as the number of wavelengths needed for rearrangeably nonblocking operation in this network.

Recall that for a general network topology, the problem of finding the optimal wavelength assignment (one which uses the minimum number of wavelengths) for a set of calls has been proven to be NP-complete [10]. In a central-switch network, however, the result turns out to be rather simple and intuitive.

Theorem 3.1 (static WA: rearrangeably nonblocking) *Assuming a priori knowledge of the entire set of call requests in a central-switch network with N stations and maximum link load L , the minimum number of wavelengths required to guarantee nonblocking operation is*

$$W_{RNB}^{static} = L \tag{3.1}$$

There are several ways to prove this result, based on different ways of representing

the central-switch network. We present a proof using the connection matrix representation, and refer the reader to Appendix A for an alternate proof using a bipartite multigraph representation.

For a central-switch network of N stations, like the one shown in Figure 2-1, we represent a set of connections and wavelength assignments using a $N \times N$ matrix whose (i, j) -th entry contains symbols representing the wavelength(s) assigned to the connection(s) from station i to station j . To satisfy the wavelength inseparability constraint, we must assign a symbol to an entry which is not already present in the row and column of that entry. This constraint, along with the maximum link load assumption, allows no more than L distinct symbols to appear in any row or column. We now prove the static WA result.

Proof of Theorem 3.1: First of all, it is clear that we need at least L wavelengths to establish a set of calls which can have L calls on a link, since each such call must use a distinct wavelength. We will present a wavelength assignment algorithm which needs *exactly* L wavelengths to satisfy any set of requests with maximum load L , thus yielding an optimal solution. This static assignment algorithm (adopted from a proof by M.C. Paull in 1961 for routing in 3-stage Clos networks [18]) assigns wavelengths to calls one-by-one *off-line*, and then backtracks when necessary to reassign calls so that no more than L wavelengths are ever used. Once every call request has been assigned a wavelength, the set of calls are then established in the network.

Assume there are exactly L wavelengths available, $\lambda_1, \lambda_2, \dots, \lambda_L$. Consider a call from station s to station d . We wish to find a wavelength (symbol) to place in the (i, j) -th entry. Without loss of generality, rearrange the rows and columns of the matrix so that the desired entry is in the upper-left corner $(1, 1)$. This does not change the wavelength assignment properties of the matrix, since the constraints for an entry only depend on the symbols in its column and row and are independent of the ordering of the columns and rows. We now claim that one of the following two scenarios must be true:

- (1) There is some wavelength (symbol), say λ_{free} , which does not appear in either

row 1 nor column 1 of the connection matrix. In this case, assigning λ_{free} to the current call satisfies the wavelength inseparability constraint, and we move onto the next call.

(2) Otherwise, if there is no such “free” wavelength, then this means that all L symbols must be present collectively in column 1 and row 1. In this case, there must be some symbol, say λ_r , in row 1 which does not appear in column 1, and a different symbol, λ_c , in column 1 which does not appear in row 1. To see this, recall that there can be at most $L - 1$ distinct symbols in row 1 and at most $L - 1$ distinct symbols in column 1. No matter how we distribute the L symbols among the entries in the row and column, there must be at least one missing symbol in the row and one missing symbol in the column. If these two symbols are the same, then we just have case (1) again.

Assuming these two symbols, λ_r and λ_c , are not the same, first rearrange the rows and columns of the matrix so that the row and column of the two symbols are adjacent to entry $(1, 1)$, as in Figure 3-1-a. Now reassign one of these two symbols to the other, say λ_c to λ_r . This reassignment may cause a conflict if there is already another λ_r in the row of the changed symbol. If so, then rearrange columns so that the conflicting λ_r , which we assume is in (i, k) , moves to column 3 (Figure 3-1-b) and reassign that symbol to λ_c . Now we must again look in column 3 for any conflicts that result from this reassignment (i.e., any other λ_c 's). We already know from our previous steps that row 1 does not contain any λ_c 's, so we only need to search those entries in column 3 which are below row 2. If there is indeed another λ_c in column 3, then we rearrange rows to move the row of that symbol to row 3 and change it to λ_r (Figure 3-1-c). Again, we search row 3 for any other λ_r 's, this time only searching entries to the right of column 2, since previous steps ensured that there are no λ_r 's in entries to the left of the changed symbol. Notice that every time we search in the row (or column) of a changed symbol, we only need to look at entries to the right (or below) that symbol.

We continue these alternating row and column rearrangements and symbol reassignments, each time searching over a smaller and smaller matrix, until we either

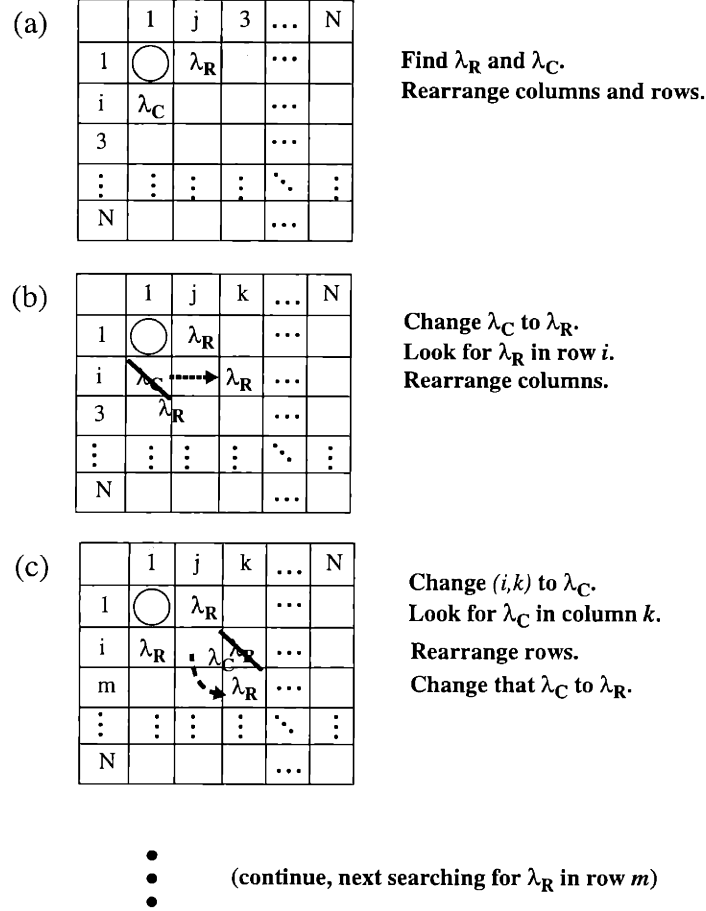


Figure 3-1: Optimal static WA algorithm

finally encounter a symbol whose re-assignment does not cause any conflicts, or until we have visited every row and column of the matrix. Since each step of the algorithm visits a new column or a new row and since the matrix has a finite number of rows and columns, the algorithm will eventually terminate. By construction, this chain of reassignments satisfies the wavelength inseparability constraint at each step, so the resulting connection matrix must also be valid.

As a result of these reassignments, row 1 and column 1 both have a common unused wavelength, in this case λ_c , which we assign to the current call. This algorithm is then repeated for the next call, and once all the requests have been assigned wavelengths, we can establish them in the network. So we have shown that as long as the arriving call requests satisfy a maximum link load L , we can find a valid wavelength assignment for every call by using no more than L wavelengths. This proves the theorem. \square

This theorem has also been presented in a more general context by Chlamtac [10]. There, it is shown that for any WDM network with an acyclic topology (assuming undirected links in the network's graph representation), the static WA problem can be solved with exactly the same number of wavelengths as needed without the wavelength continuity constraint (i.e., in a pure circuit-switched network), which is consistent with our analysis.

Theorem 3.1 implies that if the entire set of call requests in a central-switch WDM network is known beforehand, then the presence of wavelength converters in the central switch offers no benefits in terms of the minimum number of wavelengths needed for nonblocking operation.

Chapter 4

Sequential Wavelength Assignment

In the previous chapter, we assumed knowledge of the entire set of calls beforehand, and found an off-line algorithm which determines a valid set of wavelength assignments before the calls are ever established in the network. In this chapter, we assume no knowledge of future calls or disconnections and that wavelengths are assigned to calls *one at a time* with no backtracking. This constraint of not allowing previously established connections from re-tuning to different wavelengths is realistic, since high-speed optical networks can rarely afford to disrupt existing connections for fear of losing large amounts of data.

Note that if we do allow previously established calls to be reassigned wavelengths, then we can just employ the static WA algorithm of the previous chapter and use the optimal number of L wavelengths. Without this flexibility of reassignment, however, sequential WA may in general need more than L wavelengths. We begin our analysis by studying the strict-sense nonblocking condition and obtaining the minimum number of wavelengths needed under the departure scenarios of *fully-dynamic* and *semi-dynamic* calls. Then, we will study a particular class of assignment algorithms in nonblocking operation.

4.1 Strict-Sense Nonblocking

Recall that a strict-sense nonblocking network must guarantee that there will always be some valid wavelength assignment for the next call request, regardless of the previous wavelength assignments. To obtain the minimum number of wavelengths needed in this scenario, we first present a useful lemma, adopted from a study of 3-stage Clos networks by M.C. Paull in 1961 [18], which gives us the necessary conditions for requiring a new wavelength for a call.

Lemma 4.1 (necessary conditions for requiring a new wavelength) *If some call, say (i, j) , requires a new wavelength and the number of wavelengths previously used is $L + m$, then there must be at least $m + 1$ symbols in row i which are not in column j , and at least $m + 1$ symbols in column j which are not in row i .*

Proof: We define the following notation:

R = the number of symbols which are in row i but not in column j

C = the number of symbols which are in column j but not in row i

B = the number of symbols which are both in column j and row i , but not in (i, j)

X = the number of symbols which are in (i, j)

If the call (i, j) cannot use any of the previous $L + m$ wavelengths, then there must already be $L + m$ distinct symbols collectively in row i and column j . Combining this with the fact that there can be no more than L calls in row i and in column j , we have the following set of equations:

$$R + C + B + X = L + m$$

$$R + B + X < L$$

$$C + B + X < L$$

If we add m to both sides of the second and third equations and combine each of them with the first equation, we have:

$$R + C + B + X = L + m > R + B + X + m$$

which yields $C > m$, and

$$R + C + B + X = L + m > C + B + X + m$$

which yields $R > m$

Thus, if any call requires a new wavelength to be used beyond $L + m$, then there must be at least $m + 1$ symbols in the call's column which are not in its row, and at least $m + 1$ symbols in its row which are not in its column. This proves the lemma. \square

We now use this lemma to obtain the necessary conditions for strict-sense non-blocking using sequential WA.

Theorem 4.1 (sequential WA: strict-sense NB) *For sequential wavelength assignment in a central-switch network with N stations and maximum link load L , the minimum number of wavelengths required for strict-sense nonblocking operation is*

$$W_{SSNB}^{sequen} = 2L - 1 \tag{4.1}$$

Proof: We will prove this theorem in two parts, first showing that we never need more than $2L - 1$ wavelengths, and then showing that it is possible to require that many. Assume, by contradiction, that we need the $2L$ -th wavelength for call (i, j) (so that $L + m + 1 = 2L$ in the previous lemma). Then the number of symbols in row i which are not in column j must be at least $m + 1 = L$, and the same goes for the number of symbols in column j which are not in row i . But since there cannot be more than $L - 1$ symbols in either row i or column j before call (i, j) is requested, we will never need $2L$ wavelengths to establish any call. Therefore, $2L - 1$ wavelengths are sufficient to guarantee that there will always be some valid wavelength assignment for any call request.

Next, we prove that it is possible to require up to $2L - 1$ wavelengths. Consider a situation where $L - 1$ calls originate from some station i (each call on a different wavelength and all going to arbitrary destinations except to j), and $L - 1$ calls terminate at some station j (from arbitrary origins except from i). Suppose that every call into station j uses a different wavelength than any of the calls out of station i , so that there are a total of $2L - 2$ distinct wavelengths used on the two links between i and j . Now, assume that a new call from i to j arrives. We are forced to assign a completely new wavelength to this call, bringing the total number of wavelengths to $2L - 1$. This shows that it is possible to require $2L - 1$ wavelengths, thus completing the proof. \square

Since this proof was independent of the nature of disconnections of the calls, the previous strict-sense nonblocking conditions apply for both semi-dynamic and fully-dynamic traffic. The example in the second part of the proof can also be viewed in the connection matrix, as shown in Figure 4-1. The i -th row and the j -th column can collectively have, in the worst case, $(L - 1) + (L - 1)$ previously assigned symbols, each one distinct. Thus, we need at most $2L - 1$ wavelengths to establish the (i, j) -th call. This example illustrates a key property of wavelength assignment: to minimize the number of wavelengths, it is generally undesirable to have different sets of symbols in the column and row of an entry in the connection matrix. This property will be a common thread among the results to come.

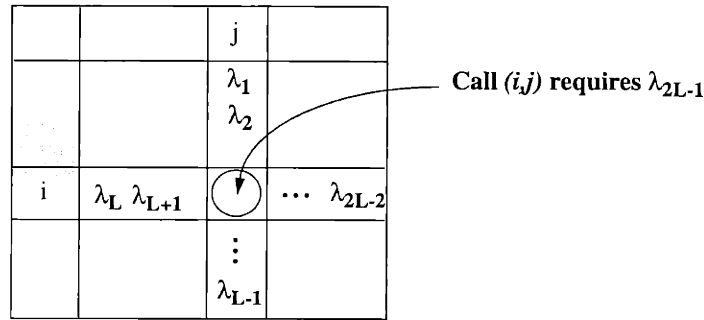


Figure 4-1: Example requiring $2L - 1$ wavelengths

We have shown that if $2L - 1$ wavelengths are available, then there will always be some valid wavelength assignment for the next call, regardless of how wavelengths were assigned to previous calls. We point out that there do exist algorithms which require up to this many wavelengths for nonblocking operation, such as *Random* assignment, described earlier, and *Least-Used* assignment [17], which assigns the next call that wavelength which has been assigned to the least number of previous calls. Note that both of these algorithms have the property of tending not to re-use wavelengths so that the distribution of assigned wavelengths becomes rather “spread out”. A natural question is whether there is some other sequential algorithm which can do better than this worst-case. Related studies of routing in 3-stage Clos networks [21, 15] have been unable to find *any* algorithm which can be proven to achieve no blocking using fewer than $2L - 1$ wavelengths in a central-switch network. In the next section, we will show that for a rather broad range of sequential algorithms, nearly $2L - 1$ wavelengths are in fact needed for no blocking.

4.2 Greedy Sequential Algorithms

We will focus on a class of sequential WA algorithms referred to as “greedy” algorithms. The name refers to the tendency of these algorithms to use the fewest number of wavelengths possible in establishing the next call. Formally, a greedy algorithm is defined as any assignment method which “never assigns a new wavelength to a call unless there is no other choice”. These algorithms are rather common in studies of scheduling and resource allocation. Several WA algorithms which have been proposed in literature fit into this category, including *First-Fit* and *Most-Used*, which assigns the next call that wavelength which is used by the most number of existing calls. We will begin by obtaining an *upper bound* on the number of wavelengths needed by greedy sequential algorithms, which follows directly from our previous strict-sense nonblocking analysis. The following theorem holds for both *fully-dynamic* and *semi-dynamic* traffic.

Theorem 4.2 (upper bound on greedy sequential WA) *For sequential wavelength assignment using a greedy algorithm in a central-switch network, the minimum number of wavelengths required for no blocking is at most*

$$W_{NB-greedy}^{sequen} \leq 2L - 1 \quad (4.2)$$

Proof: Recall from Theorem 4.1 that there will always be some valid wavelength assignment available for the next call as long as the network has $2L - 1$ wavelengths. Since greedy algorithms, by definition, never use a new wavelength unless there is no other choice, $2L - 1$ wavelengths is therefore sufficient to guarantee that greedy algorithms will never block a call. This proves the upper bound. \square

To see how many fewer wavelengths than this upper bound are possible using greedy methods, we will now obtain lower bounds on $W_{NB-greedy}^{sequen}$ for both traffic scenarios of *fully-dynamic* and *semi-dynamic* calls.

4.2.1 Fully-Dynamic Traffic

We will assume that calls are established sequentially using a greedy assignment algorithm and may then disconnect from the network at random. The following theorem gives us a lower bound on the number of wavelengths needed by any greedy algorithm to ensure no blocking in this traffic scenario.

Lemma 4.2 (fully-dynamic greedy sequential WA) *For sequential WA using a greedy algorithm for calls which can arbitrarily disconnect in a central-switch network, the minimum number of wavelengths required for nonblocking operation is at least*

$$W_{NB-greedy}^{sequen/fd} \geq \left\lfloor 2L - \frac{L}{N} \right\rfloor \quad (4.3)$$

Proof: (the following proof is adopted from a study of routing in 3-stage Clos networks by Smith [21]). We present a sequence of call arrivals and departures which forces any greedy algorithm to use the lower bound of wavelengths. First, assume each of the N stations requests L calls to itself, then the greedy rule will assign the same set of wavelengths $\lambda_1, \lambda_2, \dots, \lambda_L$ to each entry along the diagonal of the matrix. Now, for some positive integer b ($\leq L$), remove the last $L - b$ calls from station 1, and then remove the first b calls from all other stations, resulting in the matrix shown in Figure 4-2-b (for convenience, only the indices of wavelengths will be shown in matrix entries). Next, suppose station 1 requests b calls to every other station (with possibly fewer than b for the last station) until there are L calls in row 1 or until all stations have been called, whichever happens first. We see in Figure 4-2-c that each of these b calls in row 1 must be assigned a completely new set of b wavelengths.

Now we look for the value of b which forces the most number of wavelengths to be used in this example, thus yielding the tightest possible lower bound. In other words, we shall play an adversary requesting calls trying to force the network manager to assign the most number of wavelengths by choosing the appropriate b . First of all, note that for small enough b , there will not be enough entries in row 1 to reach L calls. Specifically, this happens when $b < \frac{L}{N}$. Since every additional call in row 1 must use a new wavelength, we want to choose b such that there are the maximum number of calls (L) in row 1, so clearly b must be at least as large as $\frac{L}{N}$. On the other hand, if b is too large, then the original calls in entry $(1, 1)$ will allow fewer calls in the rest of row 1, thus reducing the number of new wavelengths used. So we see that b should be just large enough to guarantee a total of L calls in row 1, but no larger than this. Thus, we choose $b = \lceil \frac{L}{N} \rceil$.

After establishing this sequence of calls, there are now L distinct symbols in row 1, all of which are different from the L symbols along the diagonals with the exception of the $b = \lceil \frac{L}{N} \rceil$ symbols in entry $(1, 1)$ (Figure 4-2-c). Adding up all the distinct

(a)		1	2	3	...	N
	1	1 ... L			...	
	2		1 ... L		...	
	3			1 ... L	...	
	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
	N				...	1 ... L

(b)		1	2	3	...	N
	1	1 ... b			...	
	2		b+1 ... L		...	
	3			b+1 ... L	...	
	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
	N				...	b+1 ... L

(c)		1	2	3	...	N
	1	1 ... b	L+1 ... L+b	L+b+1 ... L+2b	...	
	2		b+1 ... L		...	
	3			b+1 ... L	...	
	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
	N				...	b+1 ... L

Figure 4-2: Example requiring lower bound for fully-dynamic greedy sequential WA

symbols, the total number of wavelengths needed in this scenario is:

$$W = 2L - \left\lceil \frac{L}{N} \right\rceil = \left\lfloor 2L - \frac{L}{N} \right\rfloor \quad (4.4)$$

The second equality follows because for any positive integer A and positive real number x , $\lfloor A - x \rfloor = \lfloor A - \lceil x \rceil + r \rfloor = A - \lceil x \rceil$, where $r = \lceil x \rceil - x < 1$. Notice in equation 4.4 that any choice of b larger than $\lceil \frac{L}{N} \rceil$ would result in a smaller number of total wavelengths, confirming that $b = \lceil \frac{L}{N} \rceil$ maximizes the number of wavelengths needed in this example.

We have seen that in order to ensure nonblocking operation for any sequence of requests and disconnections with maximum link load L , we need at least $\lfloor 2L - \frac{L}{N} \rfloor$ wavelengths. This completes the proof of the lower bound. \square

Notice that when $L \leq N$, this lower bound simplifies to $2L - 1$, which is exactly the *upper bound* we obtained for greedy algorithms. In addition, to use k fewer wavelengths than this upper bound, a greedy method requires L to be a factor of k greater than N . If we plot the lower bound and upper bound for greedy algorithms as a function of L (Figure 4-3), we see that the two are close for most values of L and N . The potential benefit of greedy algorithms, however, does grow larger as L increases above N . This property of greedy algorithms potentially performing better at large loads will reappear again throughout our study.

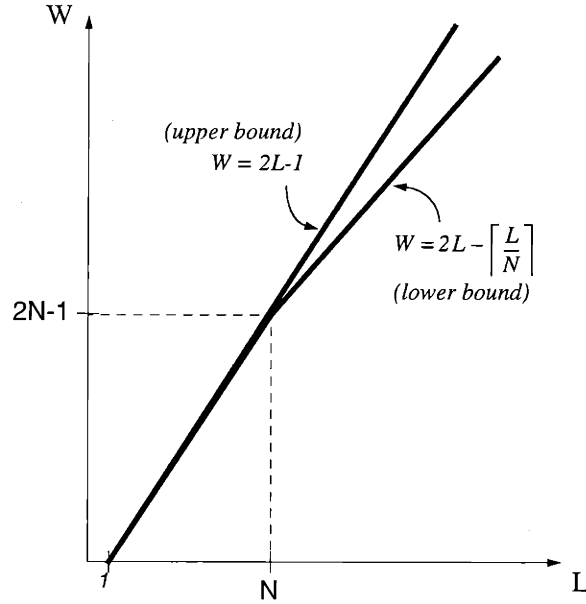


Figure 4-3: Upper and lower bounds on W_{NB} for fully-dynamic greedy sequential WA

Although this lemma has been proven for the class of greedy methods, it actually applies for a broader class of algorithms. Note that the only condition we really needed for an algorithm to satisfy the proof is that it assign the same L wavelengths to a sequence of L calls requested in each of the diagonal entries, as shown in Figure 4-2-a. Any such WA algorithm will require at least $\lceil 2L - \frac{L}{N} \rceil$ wavelengths for no blocking. In fact, the condition of Figure 4-2-a is satisfied by numerous WA algorithms proposed in literature, including: *First-Fit* [10], *Most-Used* [1], and *Max-Sum* [6]. The first two algorithms are, by definition, greedy algorithms and *Max-Sum* is greedy only if its

tie-breaking rule uses a greedy algorithm, such as *Most-Used*. All of these algorithms thus require almost $2L - 1$ wavelengths to guarantee no blocking in a central-switch network.

We now summarize our results.

Theorem 4.3 (fully-dynamic greedy sequential WA) *For greedy sequential WA for calls which can arbitrarily disconnect in a central-switch network, the minimum number of wavelengths required for no blocking is bounded by*

$$2L - 1 \geq W_{NB-greedy}^{sequen/fd} \geq \left\lfloor 2L - \frac{L}{N} \right\rfloor \quad (4.5)$$

Furthermore, when $L \leq N$, the number of wavelengths needed is exactly

$$W_{NB-greedy}^{sequen/fd} = 2L - 1 \quad (4.6)$$

Next, we consider greedy algorithms in the *semi-dynamic* case where calls may never disconnect once they are established in the network.

4.2.2 Semi-Dynamic Traffic

The traffic scenario of semi-dynamic calls has been studied before by Gerstel [13] for general WA algorithms, not limited to greedy methods. It was shown that in WDM ring networks, semi-dynamic WA requires fewer wavelengths for no blocking than fully-dynamic WA. This difference was attributed to the problem of “wavelength fragmentation” in the fully-dynamic case, where calls with shorter paths can disconnect and leave fragments of different wavelengths along a longer path, thus forcing a new call on this path to use a completely new wavelength (see [13] for a complete treatment of this topic). In particular, it was shown that for sequential WA in a ring network, the number of wavelengths required for no blocking in the fully-dynamic case is a factor of $\log N$ greater than that required in the semi-dynamic case [13].

However, we will show that in WDM central-switch networks, the difference between the two traffic cases turns out to be negligible for most values of N and L under greedy algorithms.

Lemma 4.3 (semi-dynamic greedy sequential WA) *For greedy sequential WA for calls which may never disconnect, the minimum number of wavelengths required for nonblocking operation is at least*

$$W_{NB-greedy}^{sequen/sd} \geq \left\lfloor 2L - \frac{L}{N-1} \right\rfloor \quad (4.7)$$

Proof: We present a sequence of call requests (with no departures) which requires the lower bound of wavelengths. Assume that stations 1 through $N - 1$ each requests $L - \lceil \frac{L}{N-1} \rceil$ self-calls. Any greedy algorithm will assign wavelengths $\lambda_1, \lambda_2, \dots, \lambda_{L - \lceil \frac{L}{N-1} \rceil}$ to each entry along the diagonal, as in Figure 4-4. Next, assume station N requests a total of L calls to all other stations, with a maximum of $\lceil \frac{L}{N-1} \rceil$ calls to each. Since each call in row N requires a new wavelength, there will be a total of L distinct symbols in row N , each one different from the symbols along the diagonal entries. Thus, a total of $2L - \lceil \frac{L}{N-1} \rceil = \lfloor 2L - \frac{L}{N-1} \rfloor$ wavelengths are needed in this example by any greedy algorithm, completing the proof. \square

Notice in this proof that, as in the fully-dynamic case, the only condition necessary for a WA algorithm to require this lower bound is that it assign the same set of $L - 1$ wavelengths to a sequence of $L - 1$ calls along the diagonal, as in Figure 4-4. So the previous two lower bounds on W_{NB}^{sequen} for fully-dynamic and semi-dynamic traffic in a central-switch network actually hold for a larger class of WA algorithms than just greedy methods. The reader may wonder why, then, we did not just choose our definition of greedy algorithms to be those that satisfy Figures 4-4 and 4-2. We will see in the next section that in fact the more restrictive definition of “never assign a new wavelength unless there is no other choice” is necessary in certain cases. We now summarize our results for semi-dynamic greedy algorithms.

	1	2	...	N-1	N
1	$1 \dots$ $L - \left\lceil \frac{L}{N-1} \right\rceil$...		
2		$1 \dots$ $L - \left\lceil \frac{L}{N-1} \right\rceil$...		
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
N-1			...	$1 \dots$ $L - \left\lceil \frac{L}{N-1} \right\rceil$	
N	$L - \left\lceil \frac{L}{N-1} \right\rceil + 1$ $\dots L$	$L+1 \dots$ $L + \left\lceil \frac{L}{N-1} \right\rceil$...	\dots $2L - \left\lceil \frac{L}{N-1} \right\rceil$	

Figure 4-4: Example requiring lower bound for semi-dynamic greedy sequential WA

Theorem 4.4 (semi-dynamic greedy sequential WA) *For greedy sequential WA for calls which may not disconnect from a central-switch network, the minimum number of wavelengths required for no blocking is bounded by*

$$2L - 1 \geq W_{NB-greedy}^{sequen/sd} \geq \left\lfloor 2L - \frac{L}{N-1} \right\rfloor \quad (4.8)$$

Furthermore, when $L \leq N-1$, the minimum number of wavelengths required is exactly

$$W_{NB-greedy}^{sequen/sd} = 2L - 1 \quad (4.9)$$

In summary, let us compare the lower bounds for semi-dynamic and fully-dynamic greedy algorithms with the upper bound of Theorem 4.2, as illustrated in Figure 4-5 and plotted in Figure 4-6 for different values of N . Notice that the difference between the two traffic cases is very small for most values of N and L (and they are identical when L is smaller than N). Only when L is much larger than N do greedy algorithms potentially have a significant benefit as compared to strict-sense nonblocking operation.

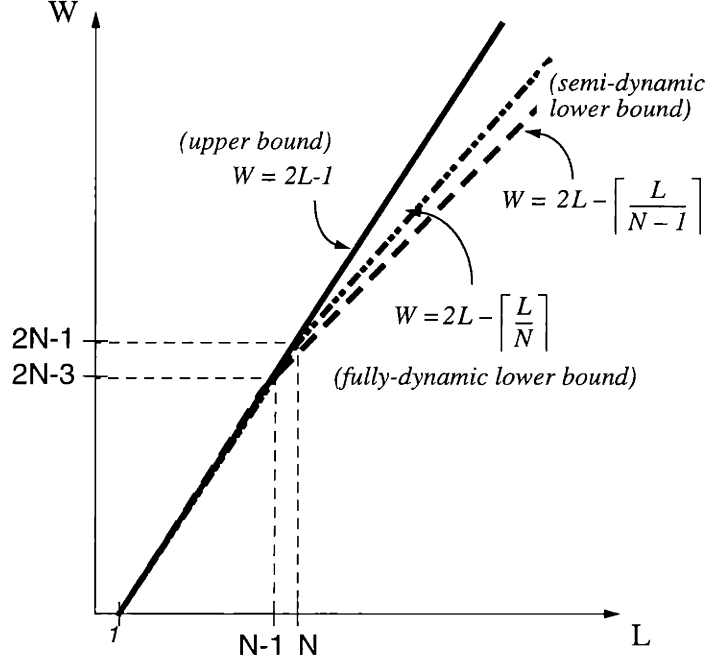


Figure 4-5: Lower and upper bounds for semi-dynamic and fully-dynamic greedy sequential WA

A natural question to ask is whether there are any non-greedy assignment algorithms which can achieve a more significant improvement. Such methods can be characterized by their tendency to assign more wavelengths than necessary to the initial few calls in hopes of using fewer wavelengths in the long run. We have already seen that *Random* and *Least-Used* assignments need exactly $2L - 1$ wavelengths. We now try to find a better non-greedy algorithm and propose one simple method which in fact requires fewer wavelengths than any greedy method for certain restricted cases. This gives us some hope that there may be other non-greedy algorithms which can *always* use fewer wavelengths than any greedy method.

4.3 A Simple Non-Greedy Algorithm

In what follows, we restrict ourselves to the assumption of *at most one directed call between any pair of source-destination stations*. This leads to the condition that $L \leq N$, since any station can be transmitting to at most N stations and receiving

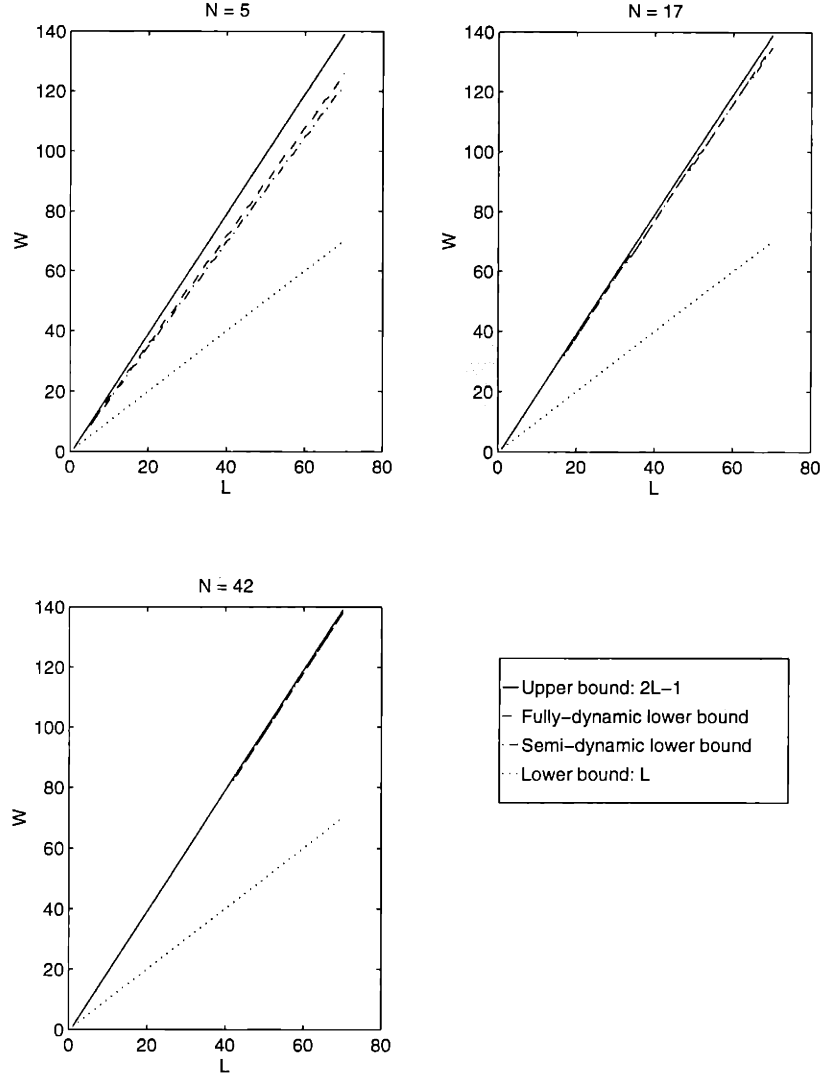


Figure 4-6: Plot of bounds for greedy sequential WA

from at most N (including a self-call). Thus, in the connection matrix, there can be at most one symbol in any entry.

We propose a simple dynamic WA algorithm where every matrix entry (connection) is pre-assigned a symbol (wavelength) such that every row is a shifted version of the row above it, as shown in Figure 4-7. This clearly satisfies the wavelength constraints and also accounts for all possible call requests, under the restriction of no repeated calls. Note that any other permutation of the rows and columns would work as well, provided no row or column has repeated symbols. For a call from i to j , we simply look up the symbol in entry (i, j) and assign the corresponding wavelength,

so we call this the “lookup table” assignment method. When L is smaller than N , we may never actually use some entries since there can be no more than L calls per row. Under the dynamic call-arrival scenario, however, we do not know *a priori* which entries will be requested, so we must pre-assign a symbol to every entry, thus using N symbols. This is independent of whether or not calls randomly leave the network.

	1	2	3	4	5
1	λ_1	λ_2	λ_3	λ_4	λ_5
2	λ_5	λ_1	λ_2	λ_3	λ_4
3	λ_4	λ_5	λ_1	λ_2	λ_3
4	λ_3	λ_4	λ_5	λ_1	λ_2
5	λ_2	λ_3	λ_4	λ_5	λ_1

Figure 4-7: Pre-assigned lookup table

Lemma 4.4 (lookup table with no repeated calls) *For the case of no repeated calls between any origin-destination pair in a central-switch network, the minimum number of wavelengths required by the “lookup table” assignment method to guarantee no blocking is*

$$W_{NB\text{-lookup}}^{seq(no\text{-}repeats)} = N \quad (4.10)$$

Proof: We have already presented most of the arguments for the proof. Again, since we assume that every entry can have at most one call, this lookup table is sufficient to satisfy any set of requests. Furthermore, since we do not have *a priori* knowledge of the calls, we need to pre-assign symbols to every entry, and this requires N wavelengths. This proves the lemma. \square

To see why we needed this restriction of no repeated calls for this method, let us try to generalize and allow multiple calls in a matrix entry. There can now be up to

L calls in any entry, so we would need NL wavelengths to account for all possible call requests by pre-assigning L symbols to every entry. Note that NL is the maximum number of possible calls which can exist simultaneously in a central-switch network, since each of the N rows and N columns in the matrix can have up to L calls. So the static lookup table is useless for general request patterns.

For comparison, we now analyze the performance of *greedy* algorithms under the assumption of no repeated calls and fully-dynamic traffic. We already know from equation 4.6 that when $L \leq N$, any greedy method needs exactly $2L - 1$ wavelengths. But the restriction of no repeated calls is an even more special case of $L \leq N$, since we now forbid more than one call per entry. Nevertheless, we will show that the result remains almost the same.

Lemma 4.5 (greedy sequential WA with no repeated calls) *For the case of no repeated calls between any origin-destination pair in a central-switch network, the minimum number of wavelengths required by any greedy sequential WA rule to guarantee no blocking is at least*

$$W_{NB-greedy}^{seq(no-repeats)} \geq 2L - 2 \quad (4.11)$$

Proof: We present a sequence of call requests satisfying the no-repeat rule and which requires $2L - 2$ wavelengths. Assume station 1 requests calls (in order) to stations $1, 2, \dots, L - 1$, with corresponding wavelengths $\lambda_1, \lambda_2, \dots, \lambda_{L-1}$ as shown in Figure 4-8-a. Next, station 2 requests calls (in order) to stations $2, 3, \dots, L - 1, 1$. By definition of the greedy rule the same set of $L - 1$ wavelengths will be assigned to these calls, in some order. Let us assume they are assigned in the same order as before, so the symbols in row 2 are a shifted version of row 1 (although any valid permutation of the $L - 1$ symbols will also work). Continue this pattern of requests for all stations up to station $L - 1$, with each station i requesting calls to stations $i, i + 1, \dots, i - 1$. Any greedy rule will reassign the same $L - 1$ symbols to each row, and we again assume without loss of generality that the assigned symbols in row i are right-shifted versions

of the row above it, shown in Figure 4-8-a. Next, each station from 1 through station $L - 1$ requests one call to station L , thus filling up column L with $L - 1$ completely new symbols, as in Figure 4-8-b. The last call in this sequence requires $2L - 2$ wavelengths, thus proving the lower bound. \square

(a)

	1	2	3	...	L-1	L	...
1	1	2	3	...	L-1		...
2	L-1	1	2	...	L-2		...
3	L-2	L-1	1	...	L-3		...
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots
L-1	2	3	4	...	1		...
L			
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots

(b)

	1	2	3	...	L-1	L	...
1	1	2	3	...	L-1	L	...
2	L-1	1	2	...	L-2	L+1	...
3	L-2	L-1	1	...	L-3	L+2	...
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots
L-1	2	3	4	...	1	2L-2	...
L			
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots

Figure 4-8: Worst-case for greedy sequential WA with no repeated calls

Since this proof did not require any disconnections, the lower bound applies to both semi-dynamic and fully-dynamic calls. If we compare the number of wavelengths required by the lookup table method versus greedy methods as a function of L , as shown in Figure 4-9, we see that the lookup table method requires fewer wavelengths for values of L in the interval $N \geq L > \frac{N}{2} + 1$. This lookup table method differs fundamentally from greedy methods in that it sacrifices immediate, or local, optimality to achieve an improved global solution. That is, for the first few call requests, the static nature of the lookup table will almost always force it to assign more wavelengths than the more dynamic greedy methods. However, as long as $N \geq L > \frac{N}{2} + 1$, we have shown that the lookup table will require fewer wavelengths in the long run.

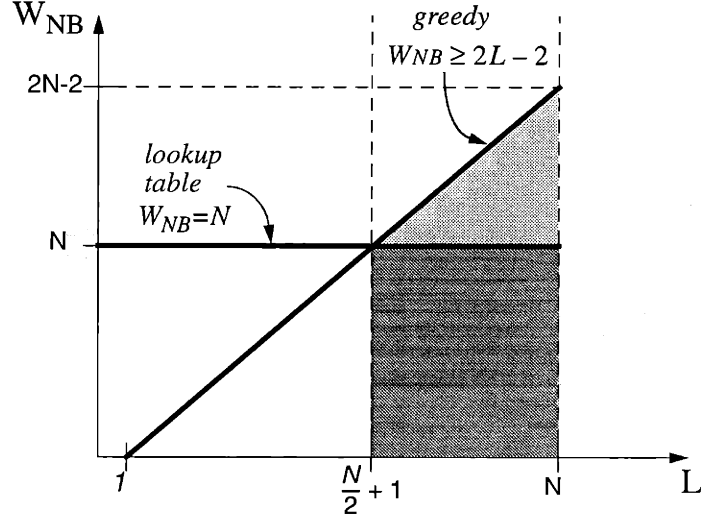


Figure 4-9: “Lookup-table” vs. greedy WA for no repeated calls

Let us now try to devise a more dynamic version of this lookup table method under the assumption of no repeated calls. The motivation for this is to use fewer wavelengths initially while still requiring N wavelengths overall. We will see, however, that such a dynamic non-greedy algorithms can be computationally difficult. Instead of pre-assigning symbols to entries, we might choose to dynamically assign wavelengths using some greedy rule, but with the constraint that each assignment must allow all entries in the matrix to have a valid assignment, with one symbol per entry. That is, for each call, we will choose a wavelength and then try to fill in the remaining empty entries using N wavelengths. If that choice of wavelength assignment does not allow the matrix to be completed, then we try subsequent wavelengths until we find one that does, and establish the call. If we can find such a wavelength for each arriving call, then this method needs N wavelengths and also has the advantage of possibly using fewer wavelengths for the first few calls. Unfortunately, the test for determining whether the matrix can be completed is equivalent to the decision problem of completing a Partial Latin Square, which was proven to be NP-complete by Colburn [12]. So even in the restricted case of no repeated calls, a more dynamic version of the lookup table is computationally difficult due to the myriad of different

possible future scenarios we must consider.

4.4 Summary of Sequential WA

We began our study of sequential WA by considering *strict-sense* nonblocking operation and showed that this requires at least $2L - 1$ wavelengths, whether or not calls are allowed to randomly disconnect. This guarantees that there will always exist some valid wavelength assignment for every call, regardless of the previous choice of assignments.

To find the potential benefits of particular algorithms, we studied the class of “greedy” algorithms, which are characterized by the property of never assigning a new wavelength unless there is no other choice, and this includes *First-Fit* and *Most-Used* algorithms. We showed that these algorithms never need more than the strict-sense nonblocking number of wavelengths. We then obtained lower bounds on the number of wavelengths needed for no blocking and saw that greedy methods in fact require almost $2L - 1$ wavelengths, but may potentially use fewer if L is larger than N . Furthermore, we pointed out that this result holds for a broader range of algorithms than just the class of greedy methods defined here, including, for example, the *Max-Sum* algorithm. The key results of our analysis can be summarized as follows:

$$W_{SSNB}^{sequen/fd} = W_{SSNB}^{sequen/sd} = 2L - 1 \geq W_{NB-greedy}^{sequen/fd} \geq W_{NB-greedy}^{sequen/sd} \geq \left\lfloor 2L - \frac{L}{N-1} \right\rfloor$$

and furthermore, $W_{NB-greedy}^{sequen/fd} \geq \left\lfloor 2L - \frac{L}{N} \right\rfloor$

We briefly considered non-greedy algorithms with a simple assignment method based on a lookup table, and showed that this requires fewer wavelengths than any greedy method under certain restricted cases. We also saw a glimpse, however, of the difficulties in devising a more general dynamic non-greedy algorithm. In short, numerous proposed sequential WA algorithms in literature require nearly $2L - 1$ wavelengths for no blocking in a central-switch network. In the next chapter, we will study the possible benefits of assigning wavelengths to batches of calls.

(**Aside:** As the proofs of this chapter indicate, there are many similarities between the problems of wavelength assignment in a central-switch WDM network and routing in a 3-stage Clos network. A more complete explanation of Clos networks and the similarities to our problem is presented in Appendix B.)

Chapter 5

Batch Wavelength Assignment

Thus far, we have shown that assigning wavelengths to calls one-by-one using a greedy algorithm requires almost twice the optimal number of wavelengths for no blocking. We are now interested in finding the possible benefits of establishing calls in *batches* at a time. Recall that sequential WA only had control over the next assignment and could not reassign previous calls. In batch WA, we now assume that we can manipulate the next *group* of assignments, but cannot reassign calls which were established in previous batches. We refer to these previous calls which are outside the current batch as “nonrearrangeable” symbols in the matrix to distinguish them from calls within the current batch which *are* rearrangeable. We will see that these nonrearrangeable symbols largely determine the number of wavelengths needed by batch WA. We begin by describing the framework and assumptions for our study.

5.1 Defining the Problem

As before, we characterize the call arrivals by assuming that no more than L calls may ever be requested on any link in the network. For batch WA, this *maximum link load* constraint applies not only for each individual group of calls, but for all connections in the network. We will further characterize each group of requests by the *minimum number of calls allowed in a batch*, designated as B . Practically, we may assume that the network manager must wait and collect the next B calls or

more before establishing them as a group in the network, and smaller batches are not allowed. Other possible characterizations include a fixed batch size and a maximum batch size. The latter is useless in our analysis since we will see that the nonblocking conditions only depend on the *smallest* batches. Fixed batch sizes may be considered as a special case of our minimum batch size assumption, but we do not study that scenario here.

One important consequence of the minimum batch size assumption is that if too many calls build up in the network, then it may prevent another batch from requesting. Specifically, recall that there can be at most NL calls in a central-switch network at any time since each of the N rows (and columns) in the connection matrix may contain at most L symbols. If the number of calls left over from previously established batches (nonrearrangeable symbols) is greater than $NL - B$ then the next batch cannot be requested until some existing connections terminate. We will see later that this characteristic of reaching the network's "saturation point" with fewer than NL calls is one factor which allows batch WA to use fewer wavelengths than sequential WA.

Using these characterizations, the fundamental question which we investigate is how the minimum number of wavelengths required for no blocking using batch WA methods depends on the minimum batch size B . As a first step, our intuition might tell us that smaller batches would in general require more wavelengths for no blocking, so that $W_{NB}^{batch}(B)$ is non-increasing with B . We prove this next.

Lemma 5.1 (smaller batches need more wavelengths for no blocking) *For batch WA with minimum batch size B , the number of wavelengths required for no blocking is a strictly non-increasing function of B . That is, for any two B_1 and B_2 such that $1 \leq B_1 \leq B_2 \leq NL$, we have*

$$W_{NB}^{batch}(B_1) \geq W_{NB}^{batch}(B_2) \tag{5.1}$$

Proof: Since we can always establish groups larger than an assumed B , any sequence

of batches with some minimum batch size specification B_2 can also be generated with a smaller minimum batch size B_1 . More rigorously, if $B_1 \leq B_2$ and we denote S_1 to be the *set of all sequences of batch requests and possible departures with minimum batch size B_1* , and S_2 to be the *set of all sequences of batch requests and possible departures with minimum batch size B_2* , then $S_2 \subseteq S_1$. Therefore, there may exist some sequences of batches generated with B_1 which cannot be generated with B_2 . In general, since some of these sequences in S_1 may require more wavelengths to be used than any sequence in S_2 , we will need *at least as many* wavelengths to establish requests with smaller B as those with larger B . This proves the lemma. \square

This lemma gives us a general picture of $W_{NB}^{batch}(B)$ dropping from a maximum at $B = 1$ down to a minimum at $B = NL$. We now make this picture more precise by first analyzing *batch strict-sense* nonblocking operation, which will have a slightly different definition than the sequential case. Then, we will obtain lower bounds on $W_{NB}^{batch}(B)$ for a particular class of batch algorithms and compare them with the strict-sense upper bound. This will allow us to determine the potential improvement offered by batch WA over sequential methods.

5.2 Strict-Sense Nonblocking

For batch WA, we will define strict-sense nonblocking as requiring the network to have enough wavelengths to ensure that there will always exist some valid *set of wavelength assignments* for the next *batch of calls*, regardless of the previous wavelength assignments. We obtain the minimum number of wavelengths required for this condition using batch WA algorithms with minimum batch size B . Our results from the previous two chapters immediately give us the endpoints of $W_{SSNB}^{batch}(B)$. For $B = 1$, we can assume that every batch consists of one call so that $2L - 1$ wavelengths are needed for strict-sense nonblocking (recall that this is independent of the nature of call departures). On the other extreme, for $B = NL$, we will show that there exists a batch WA method which needs exactly the optimal number of L wavelengths. Notice

that once the first batch of NL calls is established, the second batch must wait until every call disconnects before it can request the next NL calls (since there can be no more than NL total calls in the network at any time). Similarly, each subsequent batch of NL calls must wait until the network is empty. Since there can be at most one batch in the network at a time, we can apply an optimal static WA method, like the one in Chapter 3, to each batch and use exactly L wavelengths.

We now know that the *smallest* batches need $2L - 1$ wavelengths for strict-sense nonblocking and the *largest* batches can use exactly L wavelengths. We next obtain $W_{SSNB}^{batch}(B)$ for all other B and we will see that the result *does* depend on the nature of call departures when B is larger than 1. We begin with the fully-dynamic departure scenario, followed by the semi-dynamic and batch-dynamic cases.

5.2.1 Fully-Dynamic Traffic

In this section, we assume that calls are established in batches and may individually disconnect at random. The following theorem gives us the number of wavelengths needed for batch strict-sense nonblocking in this scenario as a function of B .

Theorem 5.1 (fully-dynamic batch WA: strict-sense NB) *For batch WA with minimum batch size B in a central-switch network with N stations and maximum link load L , if calls may individually disconnect at random then the minimum number of wavelengths required for strict-sense nonblocking is*

$$W_{SSNB}^{batch/fd}(B) = 2L - 1 \quad (5.2)$$

$$\text{for all } B \leq B_{cutoff-fd}^{SSNB}$$

and

$$W_{SSNB}^{batch/fd}(B) = L + \left\lfloor \frac{k}{2} \right\rfloor \quad (5.3)$$

$$\text{for all } B = NL - k, \text{ where } 0 \leq k \leq 2L - 2$$

where $B_{cutoff-fd}^{SSNB} = NL - 2L + 2$

This result is plotted in Figure 5-1. We now proceed to prove the theorem.

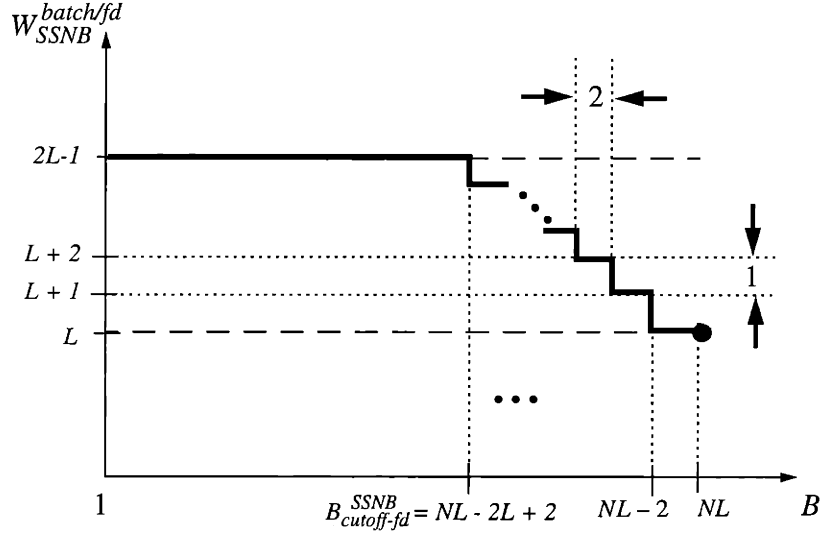


Figure 5-1: Plot of $W_{SSNB}^{batch/fd}(B)$ as a function of B

Proof: To prove this theorem, we will start with $B = NL$ and work our way down to smaller batches. Assume that $B = NL - k$ for some k such that $0 \leq k \leq NL$. In this case, there can be at most k symbols in the matrix left over from previously established batches before the next batch is established. To find $W_{SSNB}^{batch/fd}(B)$, we will take the point of view of an adversary trying to force a batch algorithm to use as many wavelengths as possible by placing the k nonrearrangeable symbols in a “worst-case” manner. Note that in the fully-dynamic case, k can be any number between 1 and NL , since we can always request NL calls and then delete any number we wish.

For a given B , our proof will consist of two parts. First we show that a certain number of wavelengths is *sufficient* to guarantee batch strict-sense nonblocking by presenting a backtracking assignment algorithm similar to the one in the proof of Theorem 3.1 which reassigns calls in the same batch to avoid using any more wavelengths. This method guarantees that the minimum number of wavelengths will be used after establishing a batch. We will then prove, for the same B , that this number

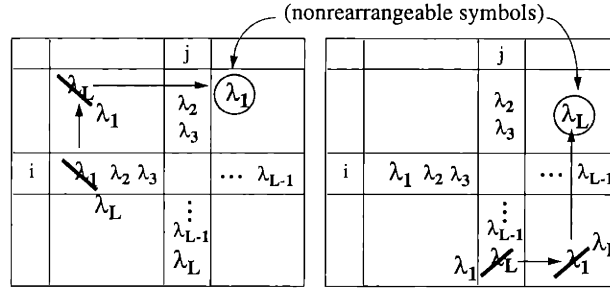
of wavelengths is *necessary* for strict-sense nonblocking by showing that we can arrange the $k = NL - B$ nonrearrangeable symbols in the matrix to satisfy the necessary conditions in Lemma 4.1 for requiring that many wavelengths.

We already know that for $k = 0$ ($B = NL$), we can always construct a chain of reassignments and never need more than L wavelengths. For a slightly smaller batch size of $B = NL - 1$, we will show that this is still true. In establishing the next batch of $NL - 1$ calls, let us assume by contradiction that some call (i, j) requires a $L + 1$ -st wavelength. Then we know from Lemma 4.1 that there must be *at least* one symbol (λ_r) in row i which does not appear in column j , and one symbol (λ_c) in column j which does not appear in row i (assume that $\lambda_r = \lambda_1$ and $\lambda_c = \lambda_L$). We will in fact assume that there is *exactly* one such symbol in the row and column, and justify this later. Furthermore, to force a batch WA to use $L + 1$ wavelengths, it cannot be possible to construct a chain of reassignments starting with either λ_1 or λ_L (otherwise (i, j) could be assigned wavelength L). This means that somewhere along the chain of reassignments for λ_1 (and for λ_L), there is a nonrearrangeable symbol which “blocks” that reassignment. If $B = NL - 1$, then there may be at most 1 call left over from previous batches (which we call λ_{NR}) and this symbol must somehow be included in *both* chains.

Since both chains only look for λ_1 in rows and λ_L in columns (Figure 5-2-a), we first assume that $\lambda_{NR} = \lambda_L$. Then both chains must look for this symbol in the same column, which means that both chains must have reassigned *the same* call with wavelength λ_1 to λ_L in the previous step in that column (since each symbol can only appear once in any row or column). Similarly, in the step before that, both chains must have reassigned the same call with wavelength λ_L to λ_1 in the same row. We continue backwards in this manner, with both chains always reassigning the same call, until we get to the first symbol of each chain. But since we assumed that both chains start from *distinct* symbols in a row and a column, we conclude that λ_{NR} cannot be λ_L . The same argument shows that λ_{NR} cannot equal λ_1 . Thus, it is not possible to block chains of reassignment for both λ_1 and λ_L using only one nonrearrangeable symbol. So if $k = 1$, we can always create chains of reassignment to avoid using

more than L wavelengths, therefore we conclude that L wavelengths is sufficient for strict-sense nonblocking for batch WA with $B = NL - 1$.

(a) Necessary conditions to require $L+1$ wavelengths for call (i,j)



(b) Forcing $L+2$ wavelengths to be used with $B = NL - 4$

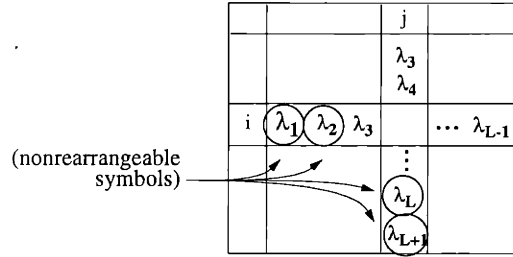


Figure 5-2: Chains of reassignment in batch WA

For the next smaller batch size, $B = NL - 2$, there can be at most two nonrearrangeable symbols. If we assume that λ_1 in row i and λ_L in column j are from previous batches, and if the next batch requests call (i, j) , then we need $L + 1$ wavelengths. We will now show that this is the most we ever need. By contradiction, let us assume that some call (i, j) in the next batch requires $L + 2$ wavelengths. Then there must be at least 2 symbols in row i which are not in column j (we assume that there are exactly two, denoted λ_r^1, λ_r^2) and at least 2 symbols in the column which are not in the row (λ_c^1, λ_c^2) and all four symbols must be *unable* to create any chain of reassignments. Since we have already shown that any two chains of reassignment starting from a symbol in a row and a symbol in a column will never share the same symbol, the only possible way to block these chains using 2 nonrearrangeable symbols is if λ_r^1 and λ_r^2 (or λ_c^1 and λ_c^2) share a common symbol somewhere in the matrix among

each of their two chains. There are two chains starting from λ_r^1 , one of them alternates looking only for λ_r^1 and λ_c^1 while the other alternates between λ_r^1 and λ_c^2 , and similarly, the two chains starting with λ_r^2 alternate between λ_r^2 and λ_c^1 and between λ_r^2 and λ_c^2 . Since there is no symbol which is common to all four chains, we cannot use fewer than 2 nonrearrangeable symbols to block every chain starting with λ_r^1 and λ_r^2 . This same argument holds for every chain starting with λ_c^1 and λ_c^2 . Therefore, we conclude that for $B = NL - 2$, we never need more than $L + 1$ wavelengths.

Similarly, for $B = NL - 3$, we can again force batch WA methods to use $L + 1$ wavelengths but never $L + 2$ since it is impossible to block every chain for 4 different calls using just 3 nonrearrangeable symbols. For $B = NL - 4$, however, we can construct an example where λ_r^1 , λ_r^2 , λ_c^1 , and λ_c^2 in row i and column j all remain from previous batches, so that the next batch will need $L + 2$ wavelengths to establish call (i, j) , as shown in Figure 5-2-b (assume that $\lambda_r^1 = \lambda_1$, $\lambda_r^2 = \lambda_2$, $\lambda_c^1 = \lambda_L$, and $\lambda_c^2 = \lambda_{L+1}$).

Notice that in order to prove *sufficiency* in our theorem, we need to find the *largest* B which requires $L + k$ wavelengths for strict-sense nonblocking. This is the reason why we assumed that there are *exactly* k symbols which are “disjoint” between row i and column j , when in fact Lemma 4.1 only requires that there be *at least* k . If we assumed that there were more than k such symbols (say $k + \Delta$), then we could only conclude that $B = NL - k - \Delta$ requires $L + k$ wavelengths. This is not sufficient to prove the theorem, since we already know that the previous examples force $L + k + \Delta$ wavelengths in this case.

So in general, with $B = NL - k$, any batch algorithm might need up to $L + \lfloor \frac{k}{2} \rfloor$ wavelengths (for $k \leq 2L - 2$), but never any more. At $k = 2L - 2$, this upper bound equals $2L - 1$, and for all larger k (so that $B < NL - 2L + 2$) we can assume that the first batch requests $2L - 2$ calls as shown in Figure 4-1, thus forcing call (i, j) in the next batch to use the $2L - 1$ -st wavelength. Thus, for all $B \leq NL - 2L + 2$, batch WA requires the same number of wavelengths as sequential WA for batch strict-sense nonblocking operation. This completes the proof. \square

We plot our result for $W_{SSNB}^{batch}(B)$ as a function of B for fully-dynamic batch WA in Figure 5-1. Notice that the range over which the number of wavelengths equals $2L - 1$ can be quite large as the number of stations grows. So unless B is very large, batch WA needs the same number of wavelengths as sequential WA to guarantee strict-sense nonblocking when calls may disconnect at random.

5.2.2 Semi-Dynamic Traffic

Now we obtain the number of wavelengths needed by batch WA for strict-sense nonblocking assuming that calls may never disconnect once they are established.

Theorem 5.2 (semi-dynamic batch WA: strict-sense NB) *For batch WA with minimum batch size B , if calls are not allowed to disconnect then the minimum number of wavelengths required for strict-sense nonblocking is*
For all $N \geq 4$,

$$W_{SSNB}^{batch/sd}(B) = 2L - 1 \tag{5.4}$$

$$\text{for all } B \leq \left\lfloor \frac{NL}{2} \right\rfloor$$

and

$$W_{SSNB}^{batch/sd}(B) = L \tag{5.5}$$

$$\text{for all } B > \left\lfloor \frac{NL}{2} \right\rfloor$$

Proof: First of all, since we assume that calls may not disconnect, if $B > \left\lfloor \frac{NL}{2} \right\rfloor$, then no further calls may be requested after the first batch is established. So we need exactly L wavelengths to establish this first batch and never any more: $W_{SSNB}^{batch/sd}(B) = L$ for all $B > \left\lfloor \frac{NL}{2} \right\rfloor$. For $B \leq \left\lfloor \frac{NL}{2} \right\rfloor$, however, there may be more than one batch in the network at a time, so we may need more than L wavelengths. We will show that exactly $2L - 1$ wavelengths are needed for strict-sense nonblocking for all values of B in this range, as long as $N \geq 4$.

Recall that a necessary condition for requiring $2L - 1$ wavelengths for batch WA is that there must be $2L - 2$ total nonrearrangeable symbols placed in the configuration shown in Figure 4-1, so that call (i, j) in the next batch requires the $2L - 1$ -st wavelength. This is always possible when $B \leq NL - 2L + 2$, which is no less than $\lfloor \frac{NL}{2} \rfloor$ for all L and for all $N \geq 4$. So under these conditions, $W_{SSNB}^{batch/sd}(B)$ equals $2L - 1$ for all B up to $\lfloor \frac{NL}{2} \rfloor$, thus completing the proof. \square

We point out that when $N = 2$ or $N = 3$ (and when L is large enough), then $NL - 2L + 2 < \lfloor \frac{NL}{2} \rfloor$, and there will be a gap between the range of B in which $W_{SSNB}^{batch/sd}(B) = 2L - 1$ and where $W_{SSNB}^{batch/sd}(B) = L$. Since fewer than $2L - 1$ wavelengths may be needed for strict-sense nonblocking in this interval of B (but never more than $2L - 1$), the step-function characteristic of the previous theorem will only provide an upper bound on $W_{SSNB}^{batch/sd}(B)$ in this gap for $N = 2$ and $N = 3$ and large L .

5.2.3 Batch-Dynamic Traffic

Now we consider the final traffic case of calls arriving and departing only in the same batches and show that the requirements for strict-sense nonblocking are the same as those in the previous semi-dynamic case.

Theorem 5.3 (batch-dynamic batch WA: strict-sense NB) *For batch WA with minimum batch size B , if calls are only allowed to disconnect in the same batches in which they were established then the minimum number of wavelengths required for strict-sense nonblocking is*

For all $N \geq 4$,

$$W_{SSNB}^{batch/bd}(B) = 2L - 1 \tag{5.6}$$

$$\text{for all } B \leq \lfloor \frac{NL}{2} \rfloor$$

and

$$W_{SSNB}^{batch/bd}(B) = L \tag{5.7}$$

for all $B > \lfloor \frac{NL}{2} \rfloor$

Proof: Since calls may only disconnect in the same batches, once B is larger than $\lfloor \frac{NL}{2} \rfloor$, then no further calls may request until the entire first batch leaves. So we see the same phenomenon as in the semi-dynamic batch WA of only one batch existing in the network at any time for this range of B , thus $W_{SSNB}^{batch/bd}(B) = L$ for all $B > \lfloor \frac{NL}{2} \rfloor$. For smaller B , we can again construct the example used in Theorem 5.2 to show that $2L - 1$ wavelengths are needed for strict-sense nonblocking for all $B \leq \lfloor \frac{NL}{2} \rfloor$, which recall is less than $NL - 2L + 2$ for all $N \geq 4$. This example does not depend on the nature of call disconnections, since all we need is to have the initial batches request $2L - 2$ calls in row i and column j with the wavelength assignments shown in Figure 4-1 and then for the second batch to request, among others, call (i, j) . So we conclude that batch WA needs the same number of wavelengths, for all B , in both the semi-dynamic and batch-dynamic cases to ensure strict-sense nonblocking for all L and all $N \geq 4$. \square

For $N = 2$ and $N = 3$ and large L , we again point out that there will be a gap between $B = NL - 2L + 2$ and $B = \lfloor \frac{NL}{2} \rfloor$, so that the step-function characteristic of $W_{SSNB}^{batch/bd}(B)$ will only be an upper bound.

5.2.4 Summary of SSNB

We have analyzed the performance of batch WA under strict-sense nonblocking operation for 3 different departure scenarios. In each of these cases, we have shown that batch WA indeed requires *fewer* wavelengths than sequential WA when B is larger than some cutoff value. For smaller B , however, batch algorithms need exactly *the same* number of wavelengths as sequential methods. We saw that this cutoff value of B can be very large in the fully-dynamic case if N is larger than L . The semi-dynamic and batch-dynamic cases generally need fewer wavelengths than fully-dynamic traffic, but this benefit was mainly due to the constraint that the network may only support one batch at a time when B is larger than $\lfloor \frac{NL}{2} \rfloor$.

Next, we study a particular class of batch WA algorithms.

5.3 Greedy Batch Algorithms

In the previous section, we saw that batch WA needs the same number of wavelengths as sequential methods for a large range of B in order to ensure strict-sense nonblocking. We now examine a weaker nonblocking condition by studying a particular class of batch algorithms. We propose a class of assignment methods called “greedy batch” algorithms. These algorithms will have the same formal definition as greedy sequential algorithms of “never assigning a new wavelength unless there is no other choice”. The difference is that now greedy batch algorithms have the flexibility to manipulate assignments for calls in the same batch to avoid using new wavelengths. As a specific illustration of this, Figure 5-3-a shows the difference between greedy sequential and one possible greedy batch algorithm. This particular greedy batch algorithm first assigns wavelengths using a greedy sequential rule, but whenever it is forced to use a new wavelength it backtracks to create one possible chain of reassignments and avoid using that new wavelength. In order to create the chain of reassignments in Figure 5-3-a, both $(2, 3)$ and $(3, 3)$ must be in the *same batch* as $(3, 2)$. If either of these calls in the chain were established in previous batches, then the chain of reassignments cannot be completed and the greedy batch method is forced to assign a new wavelength to the next call, as shown in Figure 5-3-b.

We can immediately state that the previous results on SSNB give us upper bounds for greedy batch algorithms in all three departure cases.

Theorem 5.4 (upper bound on greedy batch WA) *For greedy batch WA with minimum batch size B , the minimum number of wavelengths required for no blocking is at most*

$$W_{NB-greedy}^{batch}(B) \leq W_{SSNB}^{batch}(B) \quad (5.8)$$

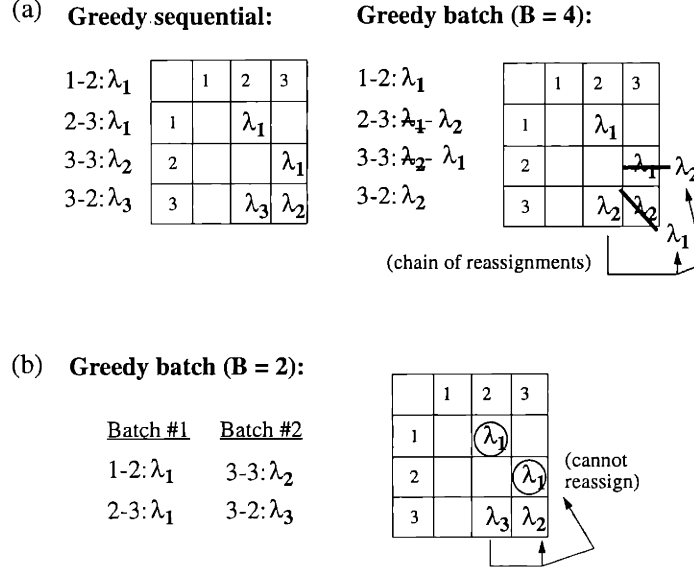


Figure 5-3: Example of greedy sequential vs. greedy batch WA

Proof: Recall that batch SSNB guarantees that there will always be a valid set of wavelengths for the next batch of requests, regardless of the previous assignments. Then greedy batch methods, by definition, will always use those available wavelengths rather than assign new ones. Thus, strict-sense nonblocking operation is sufficient to guarantee no blocking under greedy batch methods, proving the upper bound. \square

We will next obtain lower bounds on the minimum number of wavelengths needed for no blocking by greedy batch methods in all three departure scenarios.

5.3.1 Fully-Dynamic Traffic

In this section we assume that calls are established in batches using a greedy algorithm and may individually disconnect at random. We obtain a lower bound on $W_{NB-greedy}^{batch/fd}(B)$ for all B by constructing an example which forces any greedy batch algorithm to use that many wavelengths.

Theorem 5.5 (lower bound on fully-dynamic greedy batch) *For greedy batch WA with minimum batch size B , if calls may individually disconnect at random then*

the minimum number of wavelengths required for no blocking is at least
For all $N \geq 3$,

$$W_{NB-greedy}^{batch/fd}(B) \geq \left\lfloor 2L - \frac{L}{N} \right\rfloor \quad (5.9)$$

for all $B \leq B_{cutoff-fd}^{greedy}$

and

$$W_{NB-greedy}^{batch/fd}(B) \geq L + (j_c - 3) \left\lceil \frac{L}{N} \right\rceil \quad (5.10)$$

for $B_{cutoff-fd}^{greedy} < B \leq B(j_c - 1)$,

and

$$W_{NB-greedy}^{batch/fd}(B) \geq L + (j - 2) \left\lceil \frac{L}{N} \right\rceil \quad (5.11)$$

for $B(j + 1) < B \leq B(j)$,

for all integers j such that $j_c - 2 \geq j \geq 3$

and

$$W_{NB-greedy}^{batch/fd}(B) \geq L \quad (5.12)$$

for all $B > B(3)$

where $B_{cutoff-fd}^{greedy} = (N + 1 - j_c)L + 2(L - \lceil \frac{L}{N} \rceil)$

and $j_c = \left\lceil \frac{L}{\lceil \frac{L}{N} \rceil} \right\rceil + 1$

and $B(j) = (N + 1 - j)L + 2(j - 2) \lceil \frac{L}{N} \rceil$

This lower bound is sketched in Figure 5-4. Notice that as the integer j decreases, $B(j)$ increases in steps of $L - 2 \lceil \frac{L}{N} \rceil$ and $W_{NB-greedy}^{batch/fd}(B)$ decreases in steps of $\lceil \frac{L}{N} \rceil$. We now prove this result.

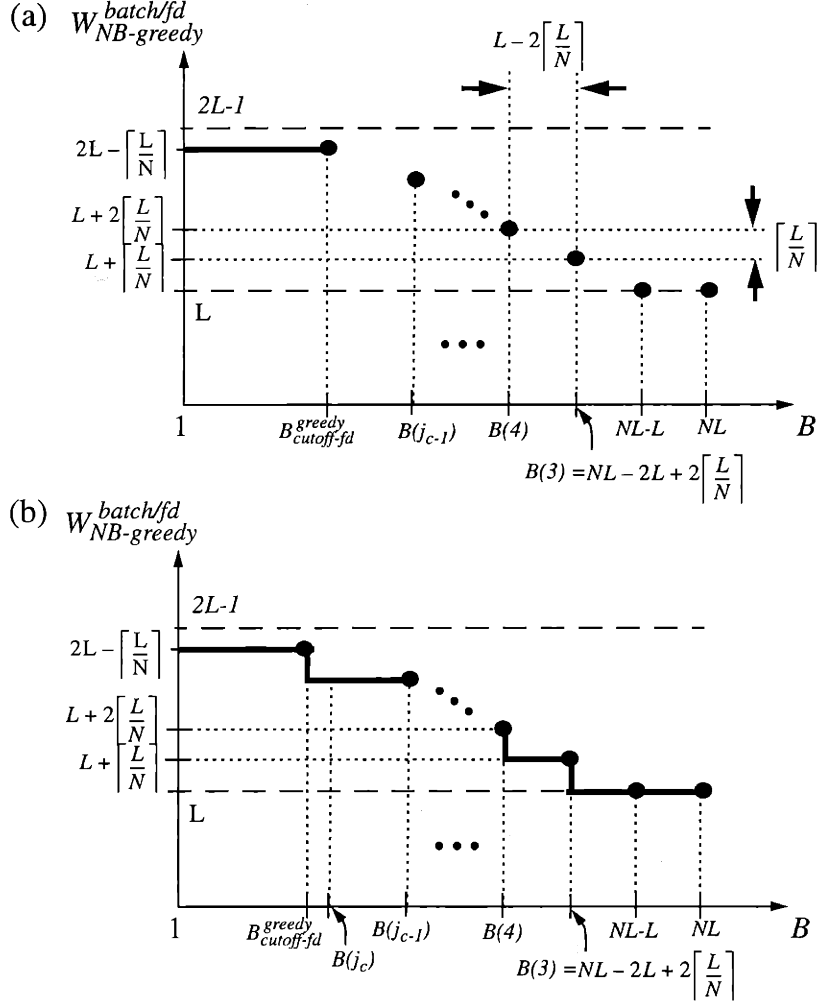


Figure 5-4: Lower bound on $W_{NB-greedy}^{batch/fd}(B)$

Proof: We prove this lower bound by finding the number of wavelengths needed by any greedy batch method for the example in Figure 5-5. This is similar to the example used to prove the fully-dynamic greedy sequential lower bound, but now there are additional calls in column 1.

We will assume without loss of generality that the first batch establishes NL calls along the diagonal, regardless of B . So the number of calls in the first batch is: $B_1 = NL$. Next, calls are removed so that only those in the first $(j-1)$ columns remain, with $\lceil \frac{L}{N} \rceil$ in entry $(1,1)$ and $L - \lceil \frac{L}{N} \rceil$ in the others (assume that $N+1 \geq j \geq 3$). Next, the second batch will start where the first batch left off and again request L

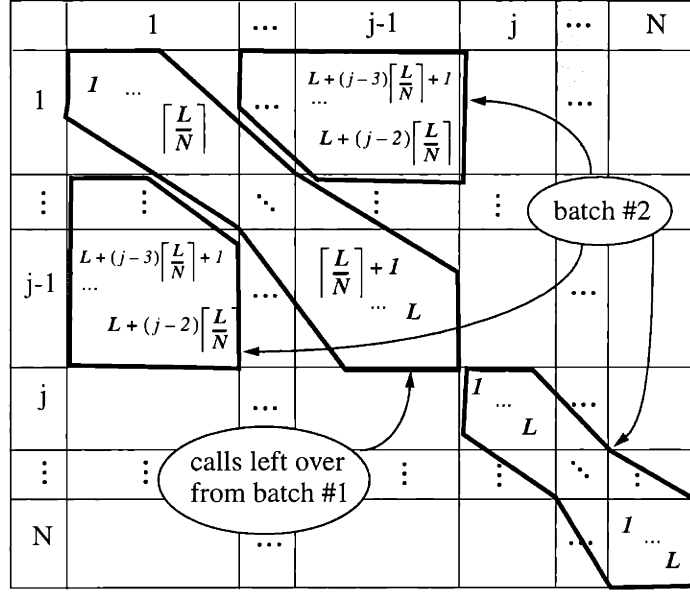


Figure 5-5: Example for the lower bound on fully-dynamic greedy batch

calls in entries (j, j) through (N, N) , followed by $\lceil \frac{L}{N} \rceil$ calls in each entry in row 1 and column 1 beginning with entries $(1, 2)$ and $(2, 1)$, assuming that $j \leq N$. For $j = N + 1$, the second batch has no calls along the diagonal and just starts in row 1 and column 1.

The second batch will stop when it reaches the $j - 1$ -st entries in row 1 and column 1, or when it reaches a total of L calls (in which case there may possibly be fewer calls than $\lceil \frac{L}{N} \rceil$ calls in the entries of the L -th calls). The completed matrix is shown in Figure 5-5 for the former case. Notice that those calls in diagonal entries to the right of column $j - 1$ may in fact be assigned *any* set of L wavelengths from 1 through the total number used in this example, and the particular choice depends on the greedy batch method used. The essential point, however, is that all the calls to the left of column j and above row j must be assigned those wavelengths shown in Figure 5-5 by any greedy batch algorithm.

Since the calls in row 1 (and column 1) may stop short of the $j - 1$ -st entry if L is small, the size of the second batch is:

$$B_2 = (N - j + 1)L + 2 \min \left\{ (j - 2) \left\lceil \frac{L}{N} \right\rceil, L - \left\lceil \frac{L}{N} \right\rceil \right\} \quad (5.13)$$

Since every call requested in row 1 and column 1 must use a new wavelength beyond L , the total number of wavelengths used after establishing the second batch is:

$$W = L + \min \left\{ (j-2) \left\lceil \frac{L}{N} \right\rceil, L - \left\lceil \frac{L}{N} \right\rceil \right\} \quad (5.14)$$

So we see that the choice of the column j in which the second batch starts determines both the number of calls in the second batch and the number of wavelengths needed. This relation between B and W will give us a lower bound on $W_{NB-greedy}^{batch/fd}(B)$. Notice that only the second batch determines the minimum batch size in this example since the first batch always requests the maximum of NL calls. To prove the first part of the lower bound, let us find the largest B which requires $\lfloor 2L - \frac{L}{N} \rfloor$ wavelengths. This “cutoff” B can be found by first obtaining the smallest j at which $(j-2) \lceil \frac{L}{N} \rceil \geq L - \lceil \frac{L}{N} \rceil$ in equation 5.14:

$$j_c = \left\lceil \frac{L + \lceil \frac{L}{N} \rceil}{\lceil \frac{L}{N} \rceil} \right\rceil = \left\lceil \frac{L}{\lceil \frac{L}{N} \rceil} \right\rceil + 1 \quad (5.15)$$

The number of calls in the second batch corresponding to j_c yields the desired “cutoff” B :

$$B_{cutoff-fd}^{greedy} = (N+1-j_c)L + 2(L - \lceil \frac{L}{N} \rceil) \quad (5.16)$$

Note that $2(L - \lceil \frac{L}{N} \rceil)$ is a *lower bound* on $B_{cutoff-fd}^{greedy}$ for all N and L . This is an equality when $j_c = N+1$, which recall means that the second batch has no calls along the diagonals and starts in row 1 and column 1. Thus, we know that $\lfloor 2L - \frac{L}{N} \rfloor$ wavelengths are always needed in this example for $B = 2(L - \lceil \frac{L}{N} \rceil)$. For values of L and N which result in $j_c = N+1$, we can say that $2(L - \lceil \frac{L}{N} \rceil)$ is also the *largest* B which needs $\lfloor 2L - \frac{L}{N} \rfloor$ wavelengths.

Due to the nonincreasing property of $W_{NB}^{batch}(B)$, we can further say that any $B \leq B_{cutoff-fd}^{greedy}$ will also require $\lfloor 2L - \frac{L}{N} \rfloor$ wavelengths for strict-sense nonblocking. This proves the first part of the lower bound.

When the second batch starts in a diagonal entry smaller than (j_c, j_c) , the number of calls in the second batch increases while the number of wavelengths decreases. Specifically, as j goes from j_c to $j_c - 1$ (so that B first exceeds $B_{cutoff-fd}^{greedy}$), the second batch adds L calls along the diagonal and loses at most $2 \lceil \frac{L}{N} \rceil$ from row 1 and column 1 (possibly fewer due to the L constraint in row 1). The number of wavelengths needed is now exactly $L + (j_c - 3) \lceil \frac{L}{N} \rceil$. Thus, for $B = (N - j_c + 2)L + 2(j_c - 3) \lceil \frac{L}{N} \rceil \equiv B(j_c - 1)$, we have that $W_{NB-greedy}^{batch/fd}(B) \geq L + (j_c - 3) \lceil \frac{L}{N} \rceil$. So now we have a range of B up to $B_{cutoff-fd}^{greedy}$ for which the lower bound is $\lfloor 2L - \frac{L}{N} \rfloor$, and a point at $B(j_c - 1)$. Since we are only looking for a lower bound, we will fill in the gap between these two points by extending the value at $B(j_c - 1)$ down to $B_{cutoff-fd}^{greedy}$. This proves the second part of the theorem.

Every subsequent decrease in j adds L more calls along the diagonal and takes away exactly $2 \lceil \frac{L}{N} \rceil$ calls in row 1 and column 1, for a net change of $L - 2 \lceil \frac{L}{N} \rceil$ calls. This expression is non-negative for all $N \geq 3$, in which case the batch size *increases* as j decreases. Meanwhile, each decrease in j leads to $\lceil \frac{L}{N} \rceil$ fewer wavelengths used. So this gives us a set of points for $W_{NB-greedy}^{batch/fd}(B)$, each corresponding to a particular j , as shown in Figure 5-4-a. Note that since we are assuming that the second batch may only start with L calls in an entry, this limits our “resolution” of the lower bound to points which are separated in B by gaps of $L - 2 \lceil \frac{L}{N} \rceil$ (with the first gap after $B_{cutoff-fd}^{greedy}$ possibly being larger). To complete the lower bound, we fill in each gap by extending the value of $W_{NB-greedy}^{batch/fd}(B)$ at the larger of the two end-points down to the smaller endpoint, thus achieving the lower bound shown in Figure 5-4-b. This proves the third part of the theorem.

Once j becomes smaller than 3, the number of wavelengths used in this example is just L . So for all $B \geq NL - 2L + 2 \lceil \frac{L}{N} \rceil$, $W_{NB-greedy}^{batch/fd} \geq L$. This completes the proof. \square

Although this is only a lower bound, it will nonetheless provide valuable insight on the potential performance of greedy batch algorithms and how this performance varies as a function of N and L . To begin with, let us analyze the “cutoff” point of B after which the lower bound first drops below $\lfloor 2L - \frac{L}{N} \rfloor$, given in equation 5.16.

First assume that L is much larger than N , so that $\frac{L}{N} \gg 1$. In this case, we can ignore the integer constraints, since $\lceil \frac{L}{N} \rceil = \frac{L}{N} + r$ is approximately equal to $\frac{L}{N}$. Then $\lceil \frac{L}{N} \rceil = N$, so the first term in equation 5.16 goes to zero, yielding $B_{cutoff-fd}^{greedy} = 2(L - \frac{L}{N})$. So when the maximum link load is much larger than the number of stations in the network, batch sizes larger than $2(L - \frac{L}{N})$ potentially allow a greedy batch algorithm to use fewer wavelengths than greedy sequential, as shown in the top part of Figure 5-6. In this plot, the upper bound equals $2L - 1 = 199$ up to $B_{cutoff-fd}^{SSNB} = 802$ (recall Theorem 5.1), at which point it drops by 1 wavelength for every increase in B by 2. The lower bound equals $\lfloor 2L - \frac{L}{N} \rfloor = 190$ up to $B_{cutoff-fd}^{greedy} = 180$, at which point it drops by 10 wavelengths for every step in B by 80.

Now assume that $L \leq N$. Then, $\lceil \frac{L}{N} \rceil = 1$, so $\lceil \frac{L}{\lceil \frac{L}{N} \rceil} \rceil = L$ and the first term in equation 5.16 is now greater than zero, so that $B_{cutoff-fd}^{greedy} = 2(L - \lceil \frac{L}{N} \rceil) + (N + 1 - L)L$. Thus, when the maximum link load is smaller than the number of stations, the range of B in which greedy batch methods need the same number of wavelengths as greedy sequential exceeds $2(L - \lceil \frac{L}{N} \rceil)$. Not only does this region become larger with decreasing L , but the lower bound in this region now *exactly equals* the upper bound of $2L - 1$, as illustrated in the lower plot of Figure 5-6. Here, the upper bound has a cutoff at $B_{cutoff-fd}^{SSNB} = 982$ and the lower bound has a cutoff at $B_{cutoff-fd}^{greedy} = 918$. Let us examine the cutoff points for the upper and lower bounds more carefully. For the upper bound, we saw that $B_{cutoff-fd}^{SSNB} = NL - (2L - 2)$. For the lower bound, equation 5.16 tells us that $B_{cutoff-fd}^{greedy} = NL - (L^2 - 2L + 2)$, since $\lceil \frac{L}{N} \rceil = 1$ when $L \leq N$. The difference between these cutoff points is $B_{cutoff-fd}^{SSNB} - B_{cutoff-fd}^{greedy} = (L - 2)^2$. So we see that the lower bound's cutoff point approaches the upper bound quadratically with decreasing L .

This characteristic of greedy batch algorithms needing exactly $2L - 1$ wavelengths for small $\frac{L}{N}$ and potentially requiring fewer wavelengths for large $\frac{L}{N}$ is reminiscent of the trend we saw in greedy sequential WA. There, recall that the lower bound for the fully-dynamic case dropped further below the upper bound as L increased, giving us hope that greedy sequential algorithms might require fewer wavelengths than required

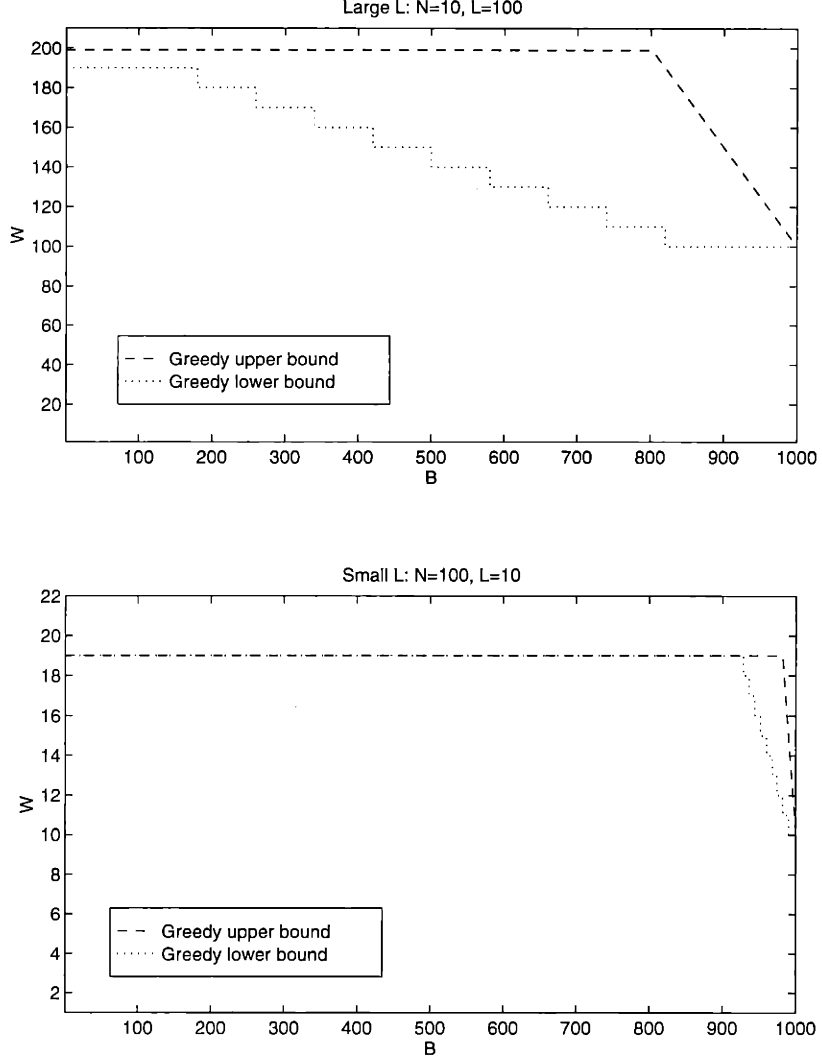


Figure 5-6: Upper and lower bounds on $W_{NB-greedy}^{batch/fd}(B)$

for SSNB. In this analysis of fully-dynamic *greedy batch*, we have seen that if $L > N$, then at least $\lfloor 2L - \frac{L}{N} \rfloor$ wavelengths are needed up to $B_{cutoff-fd}^{greedy} = 2(L - \lfloor \frac{L}{N} \rfloor)$, while for $L \leq N$, exactly $2L - 1$ wavelengths are needed up to $B_{cutoff-fd}^{greedy} = 2(L - \lfloor \frac{L}{N} \rfloor) + (N + 1 - L)L$. It may be tempting to conclude from this that $B_{cutoff-fd}^{greedy}$ is therefore larger for small $\frac{L}{N}$. But notice that these cutoff values of B increase with L , so that a direct comparison of $B_{cutoff-fd}^{greedy}$ for large L and small L is meaningless. However, we can say that in terms of batch sizes relative to the total number of possible calls in the network, larger values of $\frac{L}{N}$ allow B to be a *smaller fraction of NL* in order to achieve the same number of wavelengths for no blocking.

5.3.2 Semi-Dynamic Traffic

We might expect that the number of wavelengths needed for no blocking in the previous fully-dynamic case would be greater than the other two traffic cases, simply because we can create a much more pathological scenario if we have the flexibility of removing calls at will. So now we analyze the *semi-dynamic* scenario where calls are established in batches but may never disconnect, and obtain a lower bound on $W_{NB-greedy}^{batch/sd}(B)$ for all B .

Theorem 5.6 (lower bound on semi-dynamic greedy batch) *For greedy batch WA with minimum batch size B , if calls may not disconnect then the minimum number of wavelengths required for no blocking is at least*

$$W_{NB-greedy}^{batch/sd}(B) \geq \left\lfloor 2L - \frac{L}{N-1} \right\rfloor \quad (5.17)$$

$$\text{for all } B \leq B_{cutoff-sd}^{greedy}$$

and

$$W_{NB-greedy}^{batch/sd}(B) \geq L + (j-2) \left\lceil \frac{L}{N-1} \right\rceil \quad (5.18)$$

$$\text{for } B(j+1) < B \leq B(j),$$

$$\text{for all integers } j \text{ such that } j_c - 1 \geq j \geq j_s$$

and

$$W_{NB-greedy}^{batch/sd}(B) \geq L \quad (5.19)$$

$$\text{for all } B > B(j_s)$$

$$\text{where } B_{cutoff-sd}^{greedy} = B(j_c)$$

$$\text{and } j_c = \max \left\{ \left\lceil \frac{L}{\left\lceil \frac{L}{N-1} \right\rceil} \right\rceil + 1, j_s \right\}$$

$$\text{and } j_s = \left\lceil \frac{N+1}{2} + \frac{L}{L - \left\lceil \frac{L}{N-1} \right\rceil} \right\rceil$$

$$\text{and } B(j) = (N-j)(L - \left\lceil \frac{L}{N-1} \right\rceil) + 2L$$

This lower bound is sketched in Figure 5-7.

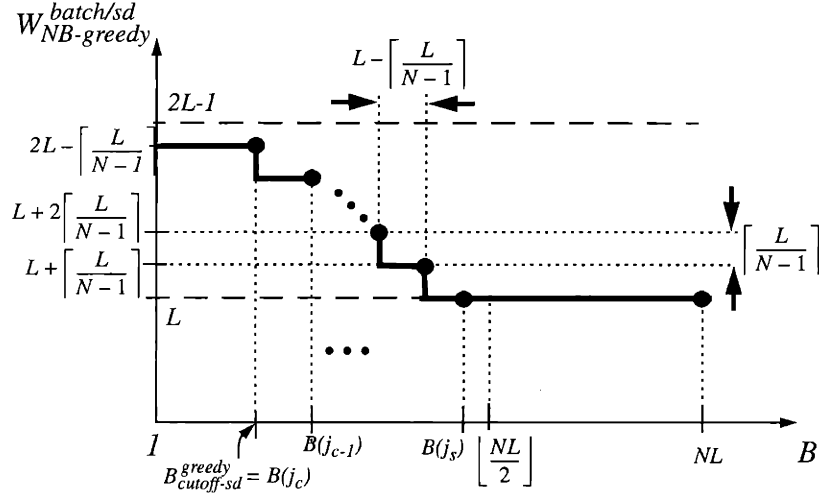


Figure 5-7: Lower bound on $W_{NB-greedy}^{batch/sd}(B)$

Proof: We prove this lower bound by finding the number wavelengths needed by a greedy batch algorithm in the example of Figure 5-8. This example is a modification of Figure 4-4 with additional calls in column N .

The first batch requests $L - \lceil \frac{L}{N-1} \rceil$ calls in each diagonal entry through column $(j-1)$, for all $N-1 \geq j \geq 2$. The total number of calls in the first batch is:

$$B_1 = (j-1)(L - \lceil \frac{L}{N-1} \rceil) \quad (5.20)$$

Next, the second batch requests $L - \lceil \frac{L}{N-1} \rceil$ calls in each entry beginning with (j, j) through $(N-1, N-1)$. It then continues and requests L calls in row N and column N , with a maximum of $\lceil \frac{L}{N-1} \rceil$ calls in each entry. We point out that for the calls to the right of column $j-1$ and below row $j-1$, Figure 5-8 shows only one possible choice of assignments, and for certain values of L and N , the second batch may not reach the last few entries in row 1 and column 1 at all. The second batch may in fact choose *any* set of $L - \lceil \frac{L}{N-1} \rceil$ valid wavelengths which are used elsewhere in the matrix. Those calls in columns 1 through $j-1$ and rows 1 through $j-1$, however, must be

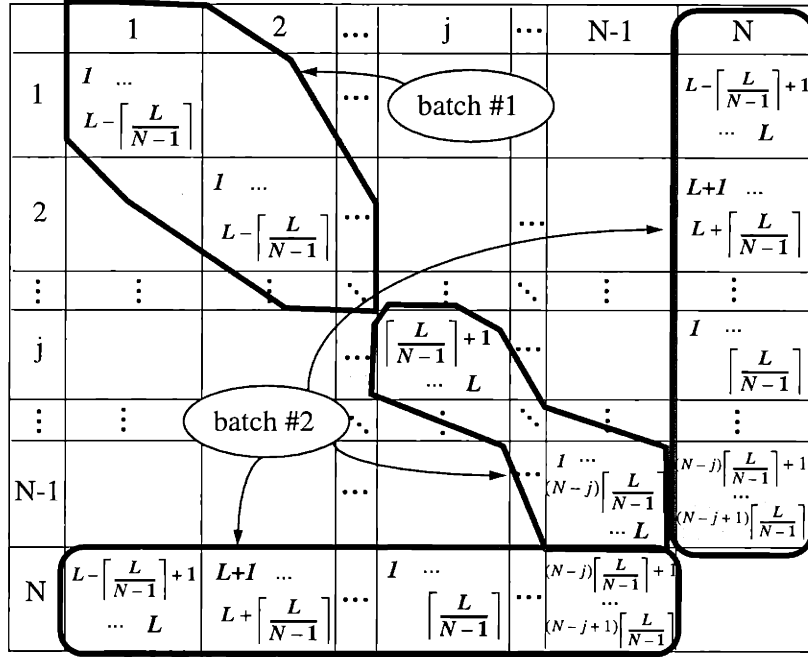


Figure 5-8: Example proving lower bound on semi-dynamic greedy batch

assigned those wavelengths shown in the figure by any greedy batch algorithm, and determine the number of wavelengths needed in this example.

The total number of calls in this second batch is:

$$B_2 = (N - j)(L - \left\lceil \frac{L}{N-1} \right\rceil) + 2L \quad (5.21)$$

The minimum of these two batch sizes will determine B in this example, so we want to make both batches as large as possible in order to get the tightest possible lower bound. The number of calls in these batches is determined by the column j in which the second batch starts, which also determines the total number of wavelengths needed, via:

$$W = L + \min \left\{ (j - 2) \left\lceil \frac{L}{N-1} \right\rceil, L - \left\lceil \frac{L}{N-1} \right\rceil \right\} \quad (5.22)$$

As j decreases, the number of calls in the second batch increases by $L - \left\lceil \frac{L}{N-1} \right\rceil$. The number of wavelengths decreases by at most $\left\lceil \frac{L}{N-1} \right\rceil$ (the last entries of row N

and column N may contain fewer). Eventually, when j is small enough, the number of calls in the second batch will exceed those in the first batch, and thus the second batch will no longer determine B . So we will restrict the second batch to start only in those diagonal entries (j, j) such that $B_1 \geq B_2$. The smallest such j which maintains this inequality is:

$$j_s = \left\lceil \frac{N+1}{2} + \frac{L}{L - \lceil \frac{L}{N-1} \rceil} \right\rceil \quad (5.23)$$

The corresponding maximum second batch size in this example is $B(j_s) \equiv (N - j_s)(L - \lceil \frac{L}{N-1} \rceil) + 2L$. It can easily be verified that this is never greater than $\lfloor \frac{NL}{2} \rfloor$, since the total number of calls in this example is at most NL (there is possibly room to add more calls in entry $(N-1, N-1)$). Since we already know that exactly L wavelengths are needed for all $B > \lfloor \frac{NL}{2} \rfloor$ in this semi-dynamic scenario, we will extend that region down to $B(j_s)$, thus obtaining the third part of the lower bound.

To prove the first part of the theorem, we find the largest range of B for which $W_{NB-greedy}^{batch/sd}(B) \geq \lfloor 2L - \frac{L}{N-1} \rfloor$. This requires that $(j-1) \lceil \frac{L}{N-1} \rceil \geq L$, so that every call in row N and column N must use a new wavelength in addition to those in the diagonals. The smallest j which ensures this is: $j_c = \left\lceil \frac{L}{\lceil \frac{L}{N-1} \rceil} \right\rceil + 1$. But since the second batch must start in diagonal entries larger than (j_s, j_s) , we must account for the possibility that j_c is smaller than j_s . So the cutoff point for the first part of the lower bound is:

$$j_c = \max \left\{ \left\lceil \frac{L}{\lceil \frac{L}{N-1} \rceil} \right\rceil + 1, j_s \right\} \quad (5.24)$$

The corresponding number of calls in the second batch gives us the cutoff B :

$$B_{cutoff-sd}^{greedy} = B(j_c) = (N - j_c)(L - \lceil \frac{L}{N-1} \rceil) + 2L \quad (5.25)$$

This proves the first part of the lower bound.

If the second batch starts in a column j between j_s and j_c , then it has $(N - j)(L - \lceil \frac{L}{N-1} \rceil) + 2L$ calls and requires $L + (j - 2) \lceil \frac{L}{N-1} \rceil$ wavelengths. This gives us a range of B for which $W_{NB-greedy}^{batch/sd}(B) \geq 2L - \lceil \frac{L}{N-1} \rceil$, and a set of points separated by $L - \lceil \frac{L}{N-1} \rceil$ calls. To obtain the complete lower bound, we fill in each gap by extending the value of W_{NB} at the upper endpoint down to the next smaller B . The complete lower bound is shown in Figure 5-7. Thus, for all B in the interval between $B(j+1) \equiv (N-j-1)(L - \lceil \frac{L}{N-1} \rceil) + 2L$ and $B(j) \equiv (N-j)(L - \lceil \frac{L}{N-1} \rceil) + 2L$, the number of wavelengths needed is at least $L + (j - 2) \lceil \frac{L}{N-1} \rceil$, where j is between j_s and j_c . This complete the proof of the theorem. \square

To study the behavior of this lower bound, we will analyze the smallest value of B at which we first need fewer than $\lfloor 2L - \frac{L}{N-1} \rfloor$ wavelengths given in equation 5.24. When L is *smaller* than N , we have $j_c = \max\{L+1, j_s\}$, where j_s now equals $\lceil \frac{N+1}{2} + \frac{L}{L-1} \rceil$. Notice that the second term in this expression is greater than 1, for all L . So if $L < \frac{N+1}{2}$, then $j_c = j_s$. This says that the second batch must use the maximum number of wavelengths for all batch sizes up to $B(j_s)$, at which point it needs just L . Thus, when L is smaller than $\frac{N+1}{2}$, the lower bound will have a sharp step characteristic at $B(j_s)$, as shown in the upper plot of Figure 5-9. Furthermore, for all B smaller than the cutoff, the lower bound of $\lfloor 2L - \frac{L}{N-1} \rfloor$ exactly equals the upper bound of $2L - 1$. In this plot, both the upper and lower bounds start with 19 wavelengths at $B = 1$ and the upper bound drops to 10 at $B = 500$ while the lower bound has the cutoff at $B = 452$.

Next, we study the case of L much larger than N , so that $\frac{L}{N-1} \gg 1$. In this case, we can ignore the integer constraints and approximate $\lceil \frac{L}{N-1} \rceil$ with $\frac{L}{N-1}$. Then we have $j_c = \max\{N, j_s\} = N$ (note that $j_s \leq N$, since otherwise $B_1 + B_2$ would exceed NL). The cutoff batch size is then: $B_{cutoff-sd}^{greedy} = 2L$. So when L is larger than N , semi-dynamic greedy batch potentially uses fewer wavelengths than greedy sequential for B as small as $2L$. Furthermore, the difference between the lower bound of $\lfloor 2L - \frac{L}{N-1} \rfloor$ and $2L - 1$ increases with $\frac{L}{N-1}$, as illustrated in the lower plot of Figure 5-9. Here, the lower bound has a cutoff at $B = 200$ and begins dropping by

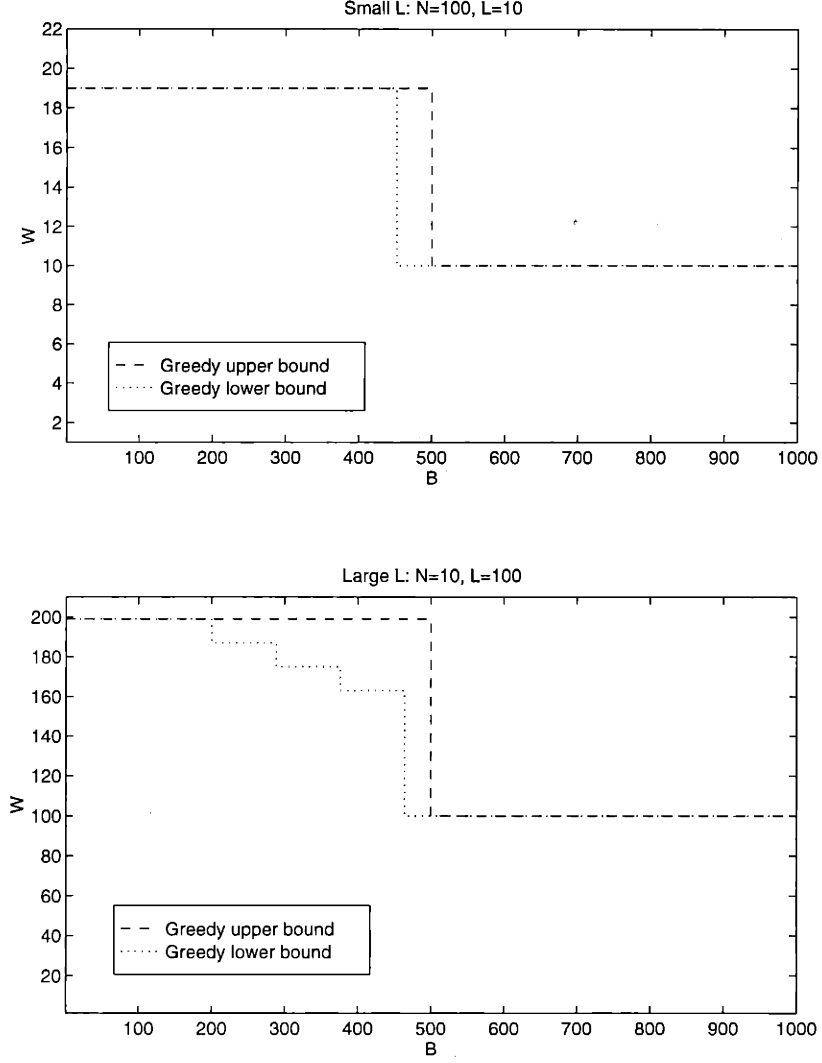


Figure 5-9: Lower bound vs. upper bound for semi-dynamic greedy batch

12 wavelengths for every increase in B by 88, until it reaches $B = B(j_s) = 464$ at which point it drops to 100 wavelengths. The two examples shown in this figure were intentionally chosen to be extreme values of N and L in order to give us a feel for the general behavior of this lower bound. Even for moderate values of N and L , however, our analysis has shown that as L grows larger than N , the lower bound falls further below the upper bound and the cutoff batch size becomes a smaller fraction of NL , similar to what we saw in the previous subsection for *fully-dynamic* greedy batch.

5.3.3 Batch-Dynamic Traffic

We now study the third possibility of *batch-dynamic* traffic, where calls are established and removed only in the same batches. We obtain a lower bound on $W_{NB-greedy}^{batch/bd}(B)$ by combining the results of the previous two theorems.

Theorem 5.7 (lower bound on batch-dynamic greedy batch) *For greedy batch WA with minimum batch size B , if calls may disconnect only in the same batches in which they were established then the minimum number of wavelengths required for no blocking is at least*

$$W_{NB-greedy}^{batch/bd}(B) \geq \left\lfloor 2L - \frac{L}{N} \right\rfloor \quad (5.26)$$

$$\text{for all } B \leq B_{cutoff-bd}^{greedy}$$

and

$$W_{NB-greedy}^{batch/bd}(B) \geq \left\lfloor 2L - \frac{L}{N-1} \right\rfloor \quad (5.27)$$

$$\text{for all } B_{cutoff-bd}^{greedy} \leq B \leq B_{cutoff-sd}^{greedy}$$

and

$$W_{NB-greedy}^{batch/bd}(B) \geq L + (j-2) \left\lceil \frac{L}{N-1} \right\rceil \quad (5.28)$$

$$\text{for } B(j+1) < B \leq B(j),$$

for all integers j such that $j_c > j \geq j_s$

and

$$W_{NB-greedy}^{batch/bd}(B) \geq L \quad (5.29)$$

$$\text{for all } B > B(j_s)$$

where $B_{cutoff-bd}^{greedy} = \min \left\{ (N-1) \left\lceil \frac{L}{N} \right\rceil, 2(L - \left\lceil \frac{L}{N} \right\rceil) \right\}$
 and $B_{cutoff-sd}^{greedy} = B(j_c)$
 and $j_c = \max \left\{ \left\lceil \frac{L}{\left\lceil \frac{L}{N-1} \right\rceil} \right\rceil + 1, j_s \right\}$
 and $j_s = \left\lceil \frac{N+1}{2} + \frac{L}{L - \left\lceil \frac{L}{N-1} \right\rceil} \right\rceil$
 and $B(j) = (N-j)(L - \left\lceil \frac{L}{N-1} \right\rceil) + 2L$

This lower bound is sketched in Figure 5-10. Notice that for small B , the lower bound is identical to that of *fully-dynamic* greedy batch, while for larger B , it is the same as the *semi-dynamic* case. We will prove the first part of this theorem by showing that for small enough B , batch-dynamic traffic needs at least the same number of wavelengths as fully-dynamic calls. Then, we will show that the rest of the lower bound can be adopted from the semi-dynamic case.

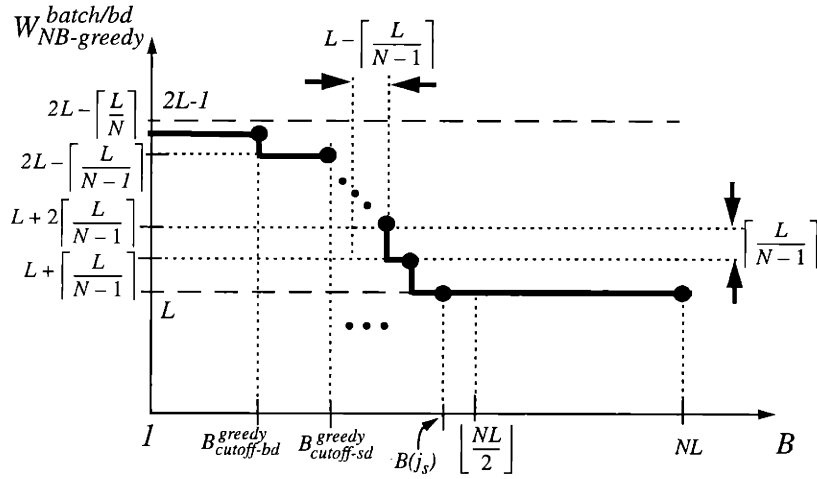


Figure 5-10: Lower bound on $W_{NB-greedy}^{batch/bd}(B)$

Proof: We prove the first part of the lower bound using a similar sequence of calls as in Theorem 5.5 for *fully-dynamic* greedy batch, but now with calls only departing in the same batches. Assume that the first batch requests $\left\lceil \frac{L}{N} \right\rceil$ calls in the diagonal entries $(2, 2)$ through (N, N) , as shown in Figure 5-11-a. The number of calls in this

batch is:

$$B_1 = (N - 1) \left\lceil \frac{L}{N} \right\rceil \quad (5.30)$$

Next, the second batch requests $\lceil \frac{L}{N} \rceil$ calls in entry $(1, 1)$ and then requests $L - \lceil \frac{L}{N} \rceil$ calls in all other diagonal entries. The number of calls in this second batch is:

$$B_2 = L + (N - 2)(L - \lceil \frac{L}{N} \rceil) \quad (5.31)$$

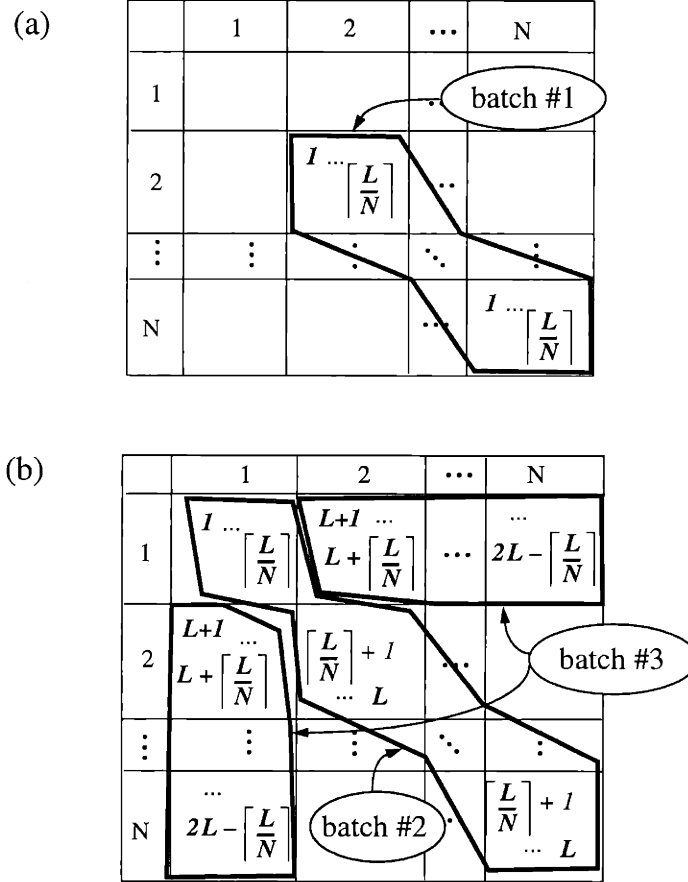


Figure 5-11: Example proving lower bound on batch-dynamic greedy batch

Next, we disconnect the first batch, leaving only the calls from the second batch along the diagonal. Then the third batch requests the remaining calls along row 1

and column 1, as shown in the figure, for a total of:

$$B_3 = 2(L - \left\lceil \frac{L}{N} \right\rceil) \quad (5.32)$$

The final matrix is shown in Figure 5-11-b. The total number of wavelengths needed for all N and L is $\lfloor 2L - \frac{L}{N} \rfloor$. The minimum batch size, B , in this example is the smallest of the three batches. Notice that $B_1 = (N-1) \left\lceil \frac{L}{N} \right\rceil = \left\lceil \frac{L}{N} \right\rceil + (N-2) \left\lceil \frac{L}{N} \right\rceil$ is always smaller than $B_2 = L + (N-2)(L - \left\lceil \frac{L}{N} \right\rceil)$ since $\left\lceil \frac{L}{N} \right\rceil < L$ and $\left\lceil \frac{L}{N} \right\rceil < L - \left\lceil \frac{L}{N} \right\rceil$ for all L and $N \geq 2$. So the second batch in this example will never determine B , thus we have: $B = \min\{B_1, B_3\}$. So for all values of B up to this cutoff, we need at least $\lfloor 2L - \frac{L}{N} \rfloor$ wavelengths. We will refer to this cutoff as $B_{cutoff-bd}^{greedy}$. This proves the first part of the theorem.

For B larger than $B_{cutoff-bd}^{greedy}$, we will just use the lower bound from the semi-dynamic greedy batch case, which must also hold under batch-dynamic traffic since we can always assume no departures. Specifically, immediately after $B_{cutoff-bd}^{greedy}$, we append the first part of the lower bound given in Theorem 5.6 which extends up to $B_{cutoff-sd}^{greedy}$. Thus, $W_{NB-greedy}^{batch/bd}(B) \geq \lfloor 2L - \frac{L}{N-1} \rfloor$ for all $B_{cutoff-bd}^{greedy} < B \leq B_{cutoff-sd}^{greedy}$. Note that the lower endpoint of this interval will never exceed the upper endpoint, since $2(L - \left\lceil \frac{L}{N} \right\rceil) < 2L \leq B_{cutoff-sd}^{greedy}$, so that $B_{cutoff-bd}^{greedy} = \min\{(N-1) \left\lceil \frac{L}{N} \right\rceil, 2(L - \left\lceil \frac{L}{N} \right\rceil)\}$, which is always smaller than $B_{cutoff-sd}^{greedy}$ for all L and N . The rest of the lower bound for B up to NL remains the same as in the semi-dynamic case. This completes the proof of the theorem. \square

Let us now examine how this lower bound varies with L and N . Since $W_{NB-greedy}^{batch/bd}(B)$ for large B is exactly the same as the semi-dynamic case studied before, we will only study the lower bound for $B \leq B_{cutoff-bd}^{greedy}$, in which case $W_{NB-greedy}^{batch/bd}(B) \geq \lfloor 2L - \frac{L}{N} \rfloor$. This region extends up to $B_{cutoff-bd}^{greedy} = \min\{(N-1) \left\lceil \frac{L}{N} \right\rceil, 2(L - \left\lceil \frac{L}{N} \right\rceil)\}$. For large $\frac{L}{N}$, we have $\frac{NL-L}{N} < 2(L - \frac{L}{N}) = 2(\frac{NL-L}{N})$. Thus, $B_{cutoff-bd}^{greedy} \approx \frac{N-1}{N}L$.

For $L \leq N$, $\left\lceil \frac{L}{N} \right\rceil = 1$, and the second term in the $\min\{\}$ expression of $B_{cutoff-bd}^{greedy}$ will be smaller than the first term as long as $L < \frac{N+1}{2}$. In this case, $B_{cutoff-bd}^{greedy} \approx$

$2(L - 1)$. Thus, relative to NL , the region of B in which greedy batch methods need the same number of wavelengths as greedy sequential becomes larger (up to almost a factor of 2) for smaller $\frac{L}{N}$. So we again see in this batch-dynamic scenario, just as in the previous two traffic cases, that the lower bound approaches the batch strict-sense nonblocking result when L is smaller than N .

5.4 Summary of Batch WA

To summarize our analysis of batch assignment methods, we compare the greedy batch lower bounds with the strict-sense nonblocking results (which are an upper bound for greedy batch) under fully-dynamic, semi-dynamic, and batch-dynamic traffic in Figure 5-12. In the uppermost plot, with $N = 60$ and $L = 10$, first notice that the fully-dynamic SSNB result (labeled “fd UB”) equals $2L - 1 = 19$ for almost all values of B , up to $B_{cutoff-fd}^{SSNB} = 582$. The lower bound for greedy batch algorithms in this fully-dynamic case, shown in the dotted line labeled “fd LB”, can do no better than batch SSNB until $B_{cutoff-fd}^{greedy} = 518$. For the other two traffic cases of semi-dynamic and batch-dynamic, both SSNB results (“sd & bd UB”) are a step function which drops to $L = 10$ at $B = 300$. The lower bounds for greedy batch algorithms in these two cases turn out to be identical in this plot (“sd & bd LB”), and are exactly the same as the SSNB curve up to $B_{cutoff-bd/sd}^{greedy} = 272$. This clearly shows us that for small values of $\frac{L}{N}$, batch algorithms can do no better than sequential methods for SSNB over a large range of B , and furthermore, that greedy batch methods need almost the same number of wavelengths as batch SSNB.

In the next two plots of Figure 5-12, we increase the ratio $\frac{L}{N}$ while keeping $NL = 600$. First of all, notice that the strict-sense NB curve for fully-dynamic traffic (“fd UB”) has a cutoff B which becomes *smaller* with increasing $\frac{L}{N}$, indicating that batch WA can use fewer wavelengths than sequential methods with smaller batch sizes. Furthermore, notice that the lower bounds for greedy batch methods under all three traffic cases drop below the SSNB curves, and that these lower bounds fall at a faster rate as $\frac{L}{N}$ increases.

We now make some comments on these plots: first of all, we approximated the fully-dynamic SSNB result (“fd UB”) by a straight line in the last two plots since the steps of $\Delta B = 2$ and $\Delta W = 1$ would not have shown up in the resolution anyway. Next, notice that the lower bounds for both semi-dynamic and batch-dynamic traffic are identical in all the plots except the third one, where the batch-dynamic lower bound is slightly larger than the semi-dynamic bound in the range $B \leq B_{cutoff-bd}^{greedy} = 54$. Finally, we point out that in the third plot, the semi-dynamic lower bound (dashed line) actually lies *above* the fully-dynamic lower bound (dotted line) for certain values of B smaller than 300. This indicates that our lower bound for fully-dynamic greedy batch WA can be improved for large $\frac{L}{N}$.

These plots show us the general trend, as L increases above N , of all three lower bounds for greedy batch algorithms falling further below the upper bounds given by batch SSNB. For small L and small B ($< \lfloor \frac{NL}{2} \rfloor$), however, all the curves are almost indistinguishable, indicating not only that there is very little difference between the three traffic cases, but that there is almost no benefit in using greedy batch assignment algorithms in central-switch networks for this range of B .

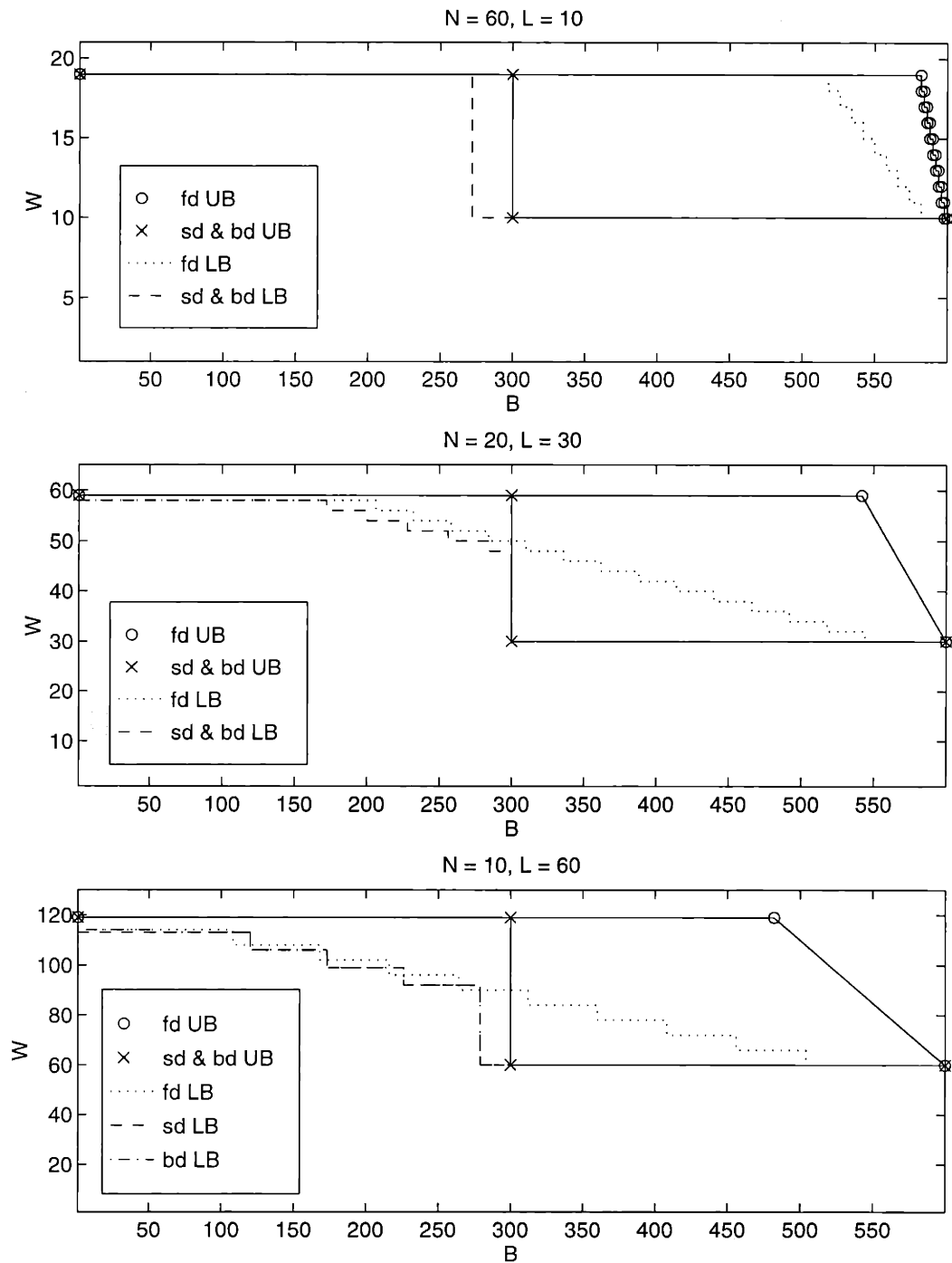


Figure 5-12: Comparison of lower bounds and upper bounds for greedy batch WA

Chapter 6

Conclusions and Further Research

We have studied the performance of both sequential and batch WA algorithms in a nonblocking central-switch network. We began with *static WA* and presented an *off-line* assignment algorithm which can use the optimal number of L wavelengths to establish any set of requests which have no more than L calls per link.

Then for *sequential WA*, where calls are established one at a time, we showed that twice the optimal number of wavelengths is needed for strict-sense nonblocking. We examined the particular class of “greedy” sequential assignment algorithms which are characterized by their preference to re-use wavelengths rather than assign new ones. We obtained lower bounds on the number of wavelengths needed, and showed that greedy algorithms need the same number of wavelengths as SSNB if $L \leq N$, and are thus no better than random WA methods. For $L \gg N$, the lower bounds for greedy methods under both fully-dynamic and semi-dynamic traffic drop below $2L - 1$, and thus only show us the best that greedy sequential methods can possibly do. We analyzed a simple non-greedy algorithm under restricted cases, and showed that it potentially uses fewer wavelengths than any greedy sequential method for a particular range of L .

We then studied the possibility of assigning wavelengths to groups of calls at a time in *batch WA*. Here, we characterized the requests by the minimum number of calls allowed in a batch, B . This led to the result that requests with larger B need *fewer wavelengths* to guarantee no blocking. We examined the performance of batch WA

under strict-sense nonblocking operation and showed that it needs the same number of $2L - 1$ wavelengths as sequential WA for values of B up to B_{cutoff}^{SSNB} , which is always at least $\lfloor \frac{NL}{2} \rfloor$. For B larger than these cutoff values, however, we have proven that batch WA needs fewer wavelengths for SSNB than sequential methods.

To find the possible benefits of using particular algorithms, we proposed the class of “greedy batch” methods, which have the same definition as greedy sequential rules of “never using a new wavelength unless there is no other choice.” We obtained lower bounds on the number of wavelengths needed for no blocking and showed that in all 3 departure scenarios, greedy batch WA needs the same number of wavelengths as greedy sequential methods when B is smaller than some cutoff B_{cutoff}^{greedy} . For B larger than these cutoffs, however, the lower bounds drop below the batch SSNB curves, thus only telling us the best that greedy batch algorithms can possibly do. Although we have only presented one possible set of lower bounds, we can nonetheless make some interesting observations on how these bounds depend on L and N , which may give us insight on how batch WA methods perform in different types of networks. In particular, we saw that as the ratio $\frac{L}{N}$ decreases, there was a convergence between the lower bounds for greedy batch methods, the batch SSNB results, and the sequential SSNB value of $2L - 1$. This indicates that batch methods offer no benefit under SSNB for a large range of B , and furthermore that the class of greedy batch algorithms can do no better. For networks with large $\frac{L}{N}$, however, we conclude that batch WA begins using fewer wavelengths than sequential methods for SSNB with smaller values of B . The lower bounds for greedy batch methods, however, are less conclusive and only tell us the best that these algorithms can possibly do.

Let us now try to interpret this dependence on L and N . For networks with $L \gg N$, there can be many calls active in the network, so that the connection matrix will be very dense. Our results would then indicate that the benefits of a “smart” WA algorithm, such as greedy sequential WA or batch WA, are accentuated in this high-congestion environment. Such algorithms may be able to avoid pathological scenarios which force other algorithms to use much more wavelengths.

For large networks with small loads, so that $L < N$, there are fewer possible calls

in the network and the connection matrix is sparse. It would seem that there are fewer opportunities for a “smart” WA algorithm to provide a benefit in this case. Indeed, our analysis supports this conjecture two-fold: first, we saw that greedy sequential algorithms need the same number of wavelengths as sequential SSNB when $L < N$, and second, we’ve also seen that the benefits of batch WA methods only begin to take effect for larger B as $\frac{L}{N}$ decreases.

In future studies, it would be interesting to determine the true performance of greedy batch algorithms for large B and large $\frac{L}{N}$, where our current lower bounds do not yield conclusive results. Furthermore, although we have seen a glimpse of sequential non-greedy algorithms, a more comprehensive study of non-greedy algorithms (for both sequential and batch WA) is needed to determine whether they can use fewer wavelengths than greedy methods.

Our basic assumption in this work was to allow no blocking. The bounds on the number of wavelengths needed were obtained by constructing bad-case scenarios. It is reasonable to ask how these bounds would change if we allowed a non-zero probability of blocking. Is the case of no blocking a “singularity” in the sense that the bad-case examples are so pathological that ignoring them will allow us to use much fewer wavelengths at very low blocking ?

Other network topologies also need to be addressed in the context of batch WA. The fact that the central-switch network is very symmetric and only allows 2-hop paths may have been a reason why batch WA methods did not show their full benefits. It is unknown whether less symmetric networks with longer maximum path lengths allow batch methods to use fewer wavelengths than sequential methods.

We have also assumed only 1 fiber on every link in the network, and it might be worthwhile to see how the addition of multiple fibers per link affects the performance of batch WA. It has been shown in a previous study that multi-fiber networks tend to amplify the benefits of “smart” sequential WA methods [6]. It seems reasonable to hope that these multi-fiber networks might also accentuate the benefits of batch WA.

Appendix A

Alternate Proof of Static WA

In addition to the connection matrix described in Chapter 3, we can also represent a central-switch network of N stations with a bipartite graph. One set of nodes represents the transmitting stations ($\leq N$) and the other set represents receiving stations ($\leq N$). We draw an edge between a pair of nodes (i, j) in this graph if there is a request from station i to station j in the network. If there are repeated requests, for example if station i requests r separate calls to station j , then we draw r edges between nodes i and j . The maximum link load of the central-switch network is equivalent to the maximum possible degree of any node in the bipartite graph.

Given this representation, assigning wavelengths to calls in the central-switch network is equivalent to edge-coloring this bipartite multigraph. To see this, note that a valid *edge-coloring* of a graph is defined to be an assignment of colors to edges such that no edges which are incident to the same node are assigned the same color. Since edges which are incident on the same node represent calls which share the same link, a valid edge-coloring satisfies both the wavelength inseparability and wavelength continuity constraints. Thus, we can find the minimum number of wavelengths needed to satisfy any set of static requests in our network by finding the minimum number of colors needed to color any set of edges in the bipartite graph. The following theorem, first presented by Konig [8], yields the desired result.

Theorem A.1 (Konig) *For a bipartite multigraph with maximum degree d , a min-*

imum edge-coloring can be achieved using d colors.

Proof: the proof is based on rearranging colors by following a chain of edges which alternates between the two groups. It is similar to the rearranging method given in the proof of Theorem 3.1, and we refer the reader to [9] for further details.

Appendix B

Routing in Clos Networks

A three-stage symmetric Clos switching network is a particular type of multistage connection network with three sets (or *stages*) of switches. Every switch in the middle stage has one link to every switch in the first and third stages. There are r switches in the first and third stages, m switches in the middle, and n ports into every first stage switch and out of every third-stage switch. To route a connection from a given input port of first-stage switch i to a desired output port of third-stage switch j , we must find an unused path through some middle-stage switch. If there is no such middle switch, then the call must be blocked.

The problem of finding a middle-stage through which to route a call in a Clos network is equivalent to the problem of finding a wavelength on which to establish a call in a WDM central-switch network. More precisely, if we let $r = N$ and $n = L$, then finding the minimum number of middle-stage switches, m , needed to ensure nonblocking operation in a Clos network is equivalent to finding the number of wavelengths needed, W , in a nonblocking WDM central-switch network.

For a complete description of routing in Clos networks and a summary of the research developments, we refer the reader to Hui [15].

Bibliography

- [1] Birman A. and Kershenbaum A. Routing and wavelength assignment methods in single-hop all-optical networks with blocking. *IEEE Infocom 1995*, page 431.
- [2] A. Aggarwal, A. Bar-Noy, et al. Efficient routing in optical networks. *Journal of the ACM*, 46(6):973–1001, November 1996.
- [3] K. Bala et al. Towards hitless reconfiguration in wdm optical networks for atm transport. *Globecom '96, London*, page 316, November 1996.
- [4] D. Banerjee and B. Mukherjee. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *IEEE JSAC*, pages 903–908, June 1996.
- [5] S. Baroni and P. Bayvel. Wavelength requirements in arbitrarily connected wavelength-routed optical networks. *IEEE JLT*, February 1997.
- [6] R.A. Barry and S. Subramaniam. The max-sum wavelength assignment algorithm for wdm ring networks. *OFC 1997*, February 1997.
- [7] V. Benes. Algebraic and topological properties of connecting networks. *Bell System Technical Journal*, July 1962.
- [8] C. Berge. *Graphs and Hypergraphs*. North Holland Publishers, 1976.
- [9] J. Carpinelli and A. Oruc. Applications of matching and edge-coloring algorithms to routing in clos networks. *Networks*, 24:319–326, 1994.

- [10] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: an approach to high-bandwidth optical wans. *IEEE Trans. Comm.*, 40(7), July 1992.
- [11] C. Clos. A study of non-blocking switching networks. *Bell System Technical Journal*, March 1953.
- [12] C.C. Colburn. The complexity of completing latin squares. *Discrete Applied Math*, 8:25–30, 1984.
- [13] O. Gerstel and S. Kutten. Dynamic wavelength allocation in all-optical ring networks. *IBM Research Report*, (RC20462), 1996.
- [14] O. Gerstel, G. Sasaki, and R. Ramaswami. Dynamic channel assignment for wdm optical networks with little or no wavelength conversion. *34th Allerton Conf. on Comm., Control, and Comp.*, 1996.
- [15] J. Hui. *Switching and Traffic Theory for Integrated Broadband Networks*. Kluwer Publishers, 1990.
- [16] M. Kovacevic and A. Acampora. Benefits of wavelength translation in all-optical clear-channel networks. *IEEE JSAC*, 14(5), 1996.
- [17] A. Mokhtar and M. Azizolglu. Adaptive wavelength routing in all-optical networks. *submitted to IEEE/ACM Trans. Networking*, 1996.
- [18] M.C. Paull. Reswitching of connection networks. *Bell System Technical Journal*, 41:833–855, May 1962.
- [19] R. Ramaswami and G. Sasaki. Multiwavelength optical networks with limited wavelength conversion. *submitted to Infocom'97*, 1996.
- [20] R. Ramaswami and K. Sivaraman. Optimal routing and wavelength assignment in all-optical networks. *Proc. IEEE INFOCOM '94*, pages 970–983, June 1994.
- [21] D.G. Smith. Lower bound on the size of a 3-stage wide-sense nonblocking network. *Electronics Letters*, 13(7):215–216, March 1997.

- [22] T.E. Stern. Linear lightwave networks. *Columbia University Technical Report*, L84-90(14), 1990.
- [23] N. Wauters and P. Demeester. Wavelength routing algorithms for transparent optical networks. *Proc. 21st ECOC'95 - Brussels*.
- [24] D.J. Wischik. Routing and wavelength assignment in optical networks. *Part III Math. Tripos, U. of Cambridge*, May 1996.