Dynamic Resource Allocation in WDM Networks with Optical Bypass and Waveband Switching

by

Li-Wei Chen

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2005

Author ....
Department of Electrical Engineering and Computer Science
July 29, 2005

Certified by ....
Prof. Eytan Modiano
Associate Professor
Thesis Supervisor

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

Dynamic Resource Allocation in WDM Networks with Optical Bypass and Waveband Switching

by

Li-Wei Chen

Submitted to the Department of Electrical Engineering and Computer Science
on July 29, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In this thesis, we investigate network architecture from the twin perspectives of link resource allocation and node complexity in WDM optical networks

Chapter 2 considers networks where the nodes have full wavelength accessibility, and investigates link resource allocation in ring networks in the form of the routing and wavelength assignment problem. In a ring network with N nodes and P calls allowed per node, we show that a necessary and sufficient lower bound on the number of wavelengths required for rearrangeably non-blocking traffic is $\lceil PN/4 \rceil$ wavelengths. Two novel algorithms are presented: one that achieves this lower bound using at most two converters per wavelength, and a second requiring $2\lceil PN/7 \rceil$ wavelengths that requires significantly fewer wavelength converters.

Chapter 3 begins our investigation of the role of reduced-complexity nodes in WDM networks by considering networks with optical bypass. The ring, torus, and tree architectures are considered. For the ring, an optical bypass architecture is constructed that requires the minimum number of locally-accessible wavelengths, with the remaining wavelengths bypassing all but a small number of hub nodes. The routing and wavelength assignment for all non-hub nodes is statically assigned, and these nodes do not require dynamic switching capability. Routing and wavelength assignment algorithms are then developed for the torus and tree architectures, and this bypass approach is extended to these topologies also.

Chapter 4 continues by considering waveband routing as a second method of reducing node complexity. We consider a two-dimensional performance space using number of wavelengths and wavebands as metrics in evaluating waveband switching networks. We derive bounds for the achievable performance region based on the minimum required number of wavelengths and wavebands. We then show by construction of several algorithms that a wavelength-waveband tradeoff frontier can be achieved that compares very favorably to the bounds.

Finally, Chapter 5 concludes by considering hybrid networks with both static and a dynamic wavelength provisioning. We use an asymptotic analysis where we allow the number of users in the network to become large, and via a geometric argument derive the optimal static and dynamic provisioning (in both wavelength-switched and waveband-switched scenarios) as a function of the traffic statistics to achieve non-blocking performance. We then extend our results to networks with a finite (possibly small) number of users where a target overflow probability is allowed. We show that by using hybrid provisioning in conjunction with waveband switching, using just a small number of switches we can obtain performance very close to a fully dynamic wavelength-switched network.

Thesis Supervisor: Prof. Eytan Modiano
Title: Associate Professor

Acknowledgements

First and foremost, I would like to express my deepest thanks to my thesis advisor, Prof. Eytan Modiano. Despite his busy schedule, he always made time for me whenever I needed it, whether I was looking for a sounding board, help with my research, a pep talk during a spell of writer's block, or advice on my career or personal life. Working with him over these last five years has been an absolute pleasure, and without him this thesis could not have been written.

I also want to thank my committee members, Prof. Vincent Chan and Prof. Muriel Medard. Their advice and constructive criticism greatly improved the quality of this thesis, especially in Chapters 4 and 5.

I greatly enjoyed working with Prof. Gregory Wornell in the inaugural offering of 6.452 this past year. His clear, simple explanations of difficult concepts is something I have striven (with varying degrees of success!) to emulate in my own writing and talks. The inspiration for the application of sphere hardening in conjunction with geometric analysis to hybrid networks in Chapter 5 struck during one of his lectures on the sphere packing proof of Shannon's capacity theorem.

I also owe a debt of gratitude to Dr. Poompat "Tengo" Saengudomlert for his help during his time as both a student and a postdoc at LIDS. Chapter 2 is complementary to his thesis work, and much of Chapter 4 was hashed out together during brainstorming sessions in our small, cozy office in Building 35 and later in our more luxurious, upscale surroundings in the Stata Center. I consider myself lucky to have counted him as both a friend and a collaborator during his stay at MIT.

I am very grateful towards all the students and staff at LIDS and MIT for fostering a fun and creative work environment. In particular, I would like to thank a few students (and former students!) individually: Shashi Borade, Andrew Brzezinski, Serena Chan, Patrick Choi, Todd Coleman, Lillian Dai, Ayres Fan, Alvin Fu, Anand Ganti, Kyle Guan, Tracey Ho, Damien Jourdan, Etty Lee, Vinson Lee, Haixia Lin, Chunmei Liu, Desmond Lun, Shubham Mukherjee, Michael Neely, Keith Santarelli, Anand Srinivas, Jun Sun, Walter Sun, Andrew Takahashi, Betty Tsai, Andy Wang, Guy Weichenberg, Yonggang Wen, and Murtaza Zafar. I also want to thank Michael Lewy, Rosangela Dos Santos, Doris Inslee, and Petr Swedock for their help over the years.

I acknowledge the support of the National Science Foundation in the research conducted in this thesis.

I would like to express my sincerest gratitude to (now Dr!) Christine H. Tsau for being a friend, a staunch supporter, a cheerleader, a confidant, and more. Thank you for sharing all the joys and sorrows of my graduate career!

Last but certainly not least, I want to thank my parents, Chao-Yuan Chen and Shing-Ming Chen, and my brother, (future Dr.) Ping-Wei Chen, for their steadfast support throughout the years. They have always encouraged me to pursue my dreams, and they believed in me even when sometimes I did not. I would not be the person I am today without their love and support, and I would like to dedicate this thesis to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optical networking has established itself as the backbone of high-speed communication systems, incorporating both high bandwidth and low noise and interference characteristics into a single medium. Within optical networks, wavelength division multiplexing (WDM) technology has emerged as an attractive solution for exploiting available fiber bandwidth to meet increasing traffic demands. WDM divides the usable bandwidth into non-overlapping frequency bands (usually referred to as *wavelengths* in the literature) and allows the same fiber to carry many signals independently by assigning each signal to a different wavelength.

WDM technology provides a number of advantages [28], including:

- **Wavelength reuse:** In a switched network, a wavelength used to carry a call in one part of the network is localized and the same wavelength can be used to carry a different call elsewhere in the network, allowing the bandwidth to be reused.

- **Transparency:** Each wavelength can carry data that is encoded and transmitted at different bit rates and using different protocols, allowing a large number of different signal types to be supported by the same network.

- **Reconfigurability:** The lightpaths provided can be set up and taken down on demand, enabling the network to support dynamically changing traffic requests.

Since the efficiency of the wavelength reuse is obviously strongly tied to the number of calls the network can support, there has been a great deal of work in the literature on the topic of efficient routing and wavelength assignment (RWA) schemes for WDM networks [2, 4, 17, 18, 29, 36, 39]. However, the RWA literature to date has largely focused on wavelength usage as the primary figure of merit. For example, [14] considers a traffic set such that the maximum load on each link is bounded by some constant, and attempts to minimize the number of wavelengths used at that given load; [26] works on minimizing the wavelength converter usage for networks using a number of wavelengths equal to the maximum link load. This assumption would be sensible if the cost of provisioning wavelengths were linear in the number of wavelengths. However, realistically the cost is usually highly non-linear. If a fiber-optic cable capable of supporting 100 wavelengths is deployed between two nodes, then the "cost" of using 70 wavelengths or 95 wavelengths on that fiber is essentially the same. (If, however, 105 wavelengths are required and a second fiber is unavailable, then suddenly a new fiber must be deployed and the cost does increase.) Add to this the large amount of dark fiber in developed environments and it quickly becomes clear that from a fiber perspective, inefficiencies in the number of wavelengths required must become large before they are costly from a network architecture perspective.

In such situations, the real cost of using large numbers of wavelengths in the network lies not in the fiber but rather in the equipment used to access the fiber. Figure 1-1 illustrates a typical node architecture for a WDM network. Each wavelength, upon arriving at a node, must be demultiplexed from the other wavelengths, switched to the appropriate output port, and then recombined with all other output wavelengths before being retransmitted. Increasing the number of wavelengths being processed at the node increases the number of demultiplexers and filters and the size of the switches required, thereby increasing the cost of the node itself. By developing efficient RWA algorithms that minimize the number of wavelengths, the nodal costs are similarly decreased.

Note that in the previous example, we have implicitly assumed a number of properties for each node, including:

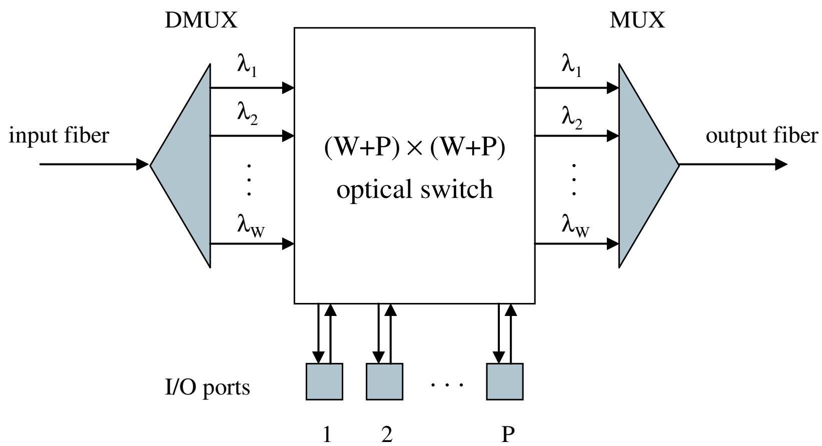


Figure 1-1: A typical node architecture for a WDM network.

- **Full reconfigurability:** All wavelengths arriving at a node are demultiplexed and actively switched. This allows for the greatest flexibility in wavelength assignment, at the cost of large switch port counts.

- **Full accessibility:** All wavelengths in the fiber enter the node. This again increases flexibility of wavelength assignment while increasing processing costs.

- **Fine grain switching:** All wavelengths are individually switched. Each wavelength must therefore have its own switch port.

Notice that each property allows for the maximum amount of freedom in routing and wavelength assignment, at the cost of greater processing requirements. In this thesis, we propose to consider relaxing some of the previous assumptions and reducing the complexity of each node. This will have the corresponding effect of introducing additional constraints into the RWA problem, and we will further investigate the impact of these additional constraints and develop new, efficient algorithms for RWA under these conditions. Using this approach, we will develop more cost-effective network architectures and achieve a greater understanding of the tradeoff between complexity and flexibility in a network.

We next discuss the relaxations we will consider in each chapter in this thesis.

1.1 Fully Flexible Node Architecture

The base case under consideration is where each wavelength is fully processed at each node. In such a scenario, we can consider the figure of merit to be solely the number of wavelengths used. Under these assumptions, we have maximum flexibility in the RWA problem, and hence the performance of an optimal RWA algorithm under these conditions provides us with a bound on the maximum efficiency of wavelength usage. This understanding will also be useful in our subsequent consideration of reduced-complexity node architectures, since it provides a basis for comparison of wavelength efficiency.

There has been considerable work done in the area of finding efficient algorithms for the RWA problem. The literature adopts a number of different approaches to modeling the problem. In the static traffic model, the traffic matrix representing the calls is fixed and does not change over time. In the dynamic traffic model, the traffic matrix is allowed to change over time to represent call arrivals and departures.

In the static model, the objective is typically to minimize the number of wavelengths, converters, or other cost parameters [28]. This problem was shown to be NP-complete in [8], and thus the literature has focused on the development of heuristics and bounds. Other approaches include attempting to maximize throughput for fixed capacity [1], to minimize congestion for a fixed traffic set [19], or to maximize the number of calls supported for a fixed number of wavelengths [27]. However, this approach is limited in that it does not allow dynamic call setup and removals. Additionally, if the traffic is static, nodal connections can be hard-wired to make for "dumb" nodes without reconfigurability, and a flexible node architecture is not required.

The alternative is to use a dynamic model, where calls are allowed to arrive and depart over time. Since we are interested in reconfigurable nodes, we will primarily be interested in this dynamic traffic scenario, where we can use the reconfigurability to accommodate changes in the traffic. One method of modeling call dynamics is to adopt a statistical model for call arrival rates and holding times and design algorithms to minimize the call blocking probability. Numerous papers have focused on blocking probability analysis under various approximations for simple wavelength assignment algorithms such as the random algorithm [4, 2, 18, 36, 39, 29] and first-fit [17]. However, due to the large state-space size of the problem, the blocking probability of a WDM network for more sophisticated algorithms is extremely difficult to analyze. As a result, most statistical algorithms rely on simplifying approximations and heuristics [21].

An alternative approach considers designing the network to accommodate any traffic matrix from an admissible set. Call arrivals or departures are equivalent to transitioning from one traffic matrix to another. If the transitions can be accommodated without rearranging any calls, the RWA algorithm is called *wide-sense non-blocking*; algorithms which require call rearrangement are called *rearrangeably non-blocking*. For example, [14] considers a traffic set such that the maximum load on each link is bounded by some constant, and attempts to minimize the number of wavelengths used at that given load; [26] works on minimizing the wavelength converter usage for networks using a number of wavelengths equal to the maximum link load. Another approach is taken in [24] by admitting any traffic matrix where each node uses at most P ports. It is shown that for the case of a bidirectional ring with N nodes and P ports, a lower bound of $\lceil PN/3 \rceil$ wavelengths is required to support the worst-case traffic set if no wavelength conversion is employed. Moreover, in [24] a rearrangeably non-blocking RWA algorithm is provided which achieved this bound. An online version based on these ideas was presented in [31] which additionally attempts to minimize the number of rearrangements required; this algorithm was later extended from rings to torus networks in [33]. The P-port model is practical since the admissible set is based on actual device limitations in the network. In Chapter 2, we will investigate new rearrangeably non-blocking RWA schemes using the P-port model where wavelength conversion is available.

We will consider a simple yet commonly-used symmetric network architecture, the ring network. We will develop theoretical bounds on the minimum number of wavelengths required to support rearrageably non-blocking traffic given full wavelength conversion, and show that a necessary and sufficient number of wavelengths to support P-port traffic in such a network is $\lceil PN/4 \rceil$. We will also show by construction that this bound is achievable using a minimal number of wavelength converters. A second algorithm using fewer converters but more wavelengths will also be given. This chapter will also investigate the issue of the placement of wavelength converters and give lemmas describing how to design to allow arbitrary placement of converters within the network.

1.2 Optical Bypass - Partial Wavelength Accessibility

Chapter 2 considered routing and wavelength assignment, and used total number of wavelengths as the cost metric. This is sensible if all wavelengths are accessed at all nodes – reducing the number of wavelengths therefore reduces the hardware costs as well.

However, in a medium to large network, the amount of traffic (in wavelengths) that each node contributes is typically a small fraction of the total number of wavelengths in the fiber. It therefore makes sense to consider allowing the majority of the wavelengths entering a node to simply optically bypass the node, and tap only a fraction of the wavelengths that is sufficiently large for adding outgoing calls and removing incoming ones. So-called bypass wavelengths then become very inexpensive to manage from a hardware perspective, since they will only be processed at a very limited subset of access nodes. Under this architecture, we impose a limited-accessibility constraint to the wavelength set (making the RWA problem more complex) in order to reduce overall hardware complexity.

We will refer to *banding* as the grouping of wavelengths into frequency bands; each band contains multiple adjacent wavelengths. As mentioned, node complexity can be significantly reduced by allowing some bands to completely bypass each node. This is permissible if the RWA algorithm can guarantee that wavelengths within the

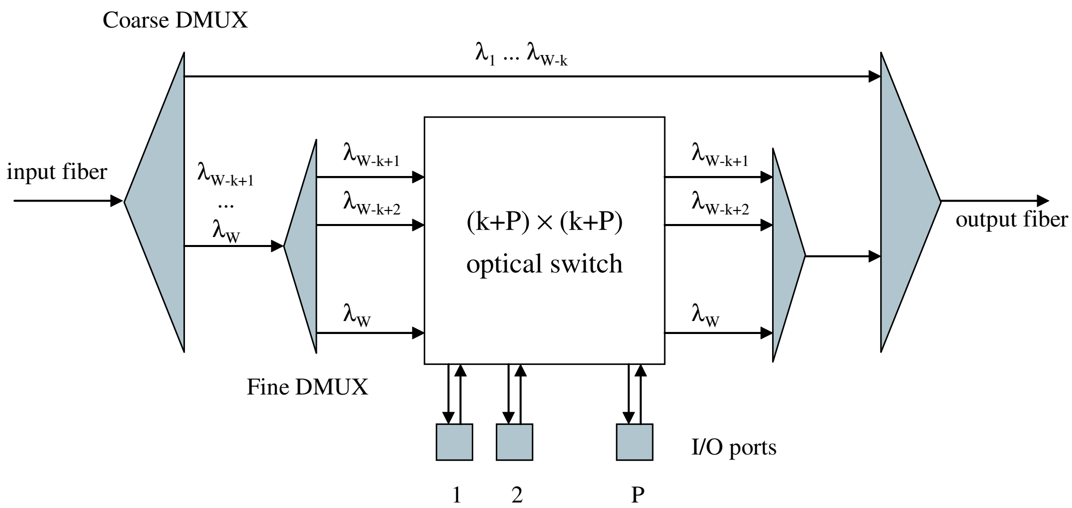


Figure 1-2: An OADM architecture with bypass. Note that the switch size depends only on the number of wavelengths k entering the node, not the total number W.

bypassing bands are never dropped at those nodes. For the purposes of the RWA problem, we can group the wavelengths into two bands: a *local* band, consisting of wavelengths that can be accessed by all nodes, and a *bypass* band, consisting of wavelengths that can be accessed only by a few designated *hub nodes*. The bypass band can therefore bypass the majority of the nodes in the network. There are several advantages to a banding approach. One is cost savings. Figure 1-2 shows an optical add-drop multiplexer (OADM) in a system where the total number of wavelengths W are divided into a local band of k wavelengths and a bypass band of $W - k$ wavelengths. With banding, only the smaller local band of k wavelengths is switched. Without banding, the switch would have to process all W wavelengths. Another benefit is that the wavelength demultiplexers can be simpler: the first, coarse DMUX need only separate out two large bands, while the second, finer DMUX has a smaller band to work with (only the local wavelengths). Finally, by allowing wavelengths in the bypass band to avoid processing at non-hub nodes altogether, the bypass band can either avoid the switch (thus not suffering power losses due to switching which would reduce the reach of the lightpaths), or be placed in a separate fiber entirely. Such a separate fiber would need to be connected only to the hub nodes and could physically bypass all other nodes entirely.

In Chapter 3, this optical bypass problem is investigated. Specifically, we focus on locally-accessible wavelengths as the primary cost criterion, and determining the minimum number of locally-accessible wavelengths required. Total number of wavelengths are also a (secondary) consideration, and we investigate architectures that allow the majority of wavelengths to optically bypass most nodes. Three topologies are considered: rings, toruses, and trees.

For rings, we will begin with the results of [30] and Chapter 2 for routing and wavelength assignment with and without converters. We develop a novel approach for banding that minimizes the number of locally-accessible wavelengths, and additionally eliminates the need for switching for all but a designated small number of so-called "hub nodes".

For toruses, we derive a lower bound on the number of wavelengths required to support P-port traffic, and provide an algorithm which performs favorably compared to this bound. We then extend the banding approach from the ring topology to the torus as well.

For trees, we again derive a lower bound on the number of wavelengths required, and use a ring-embedding approach to develop a low-complexity algorithm that performs well compared to this bound. The banding approach is then further extended to the tree topology.

1.3 Multigranularity Switching

In Chapter 4, we propose a second option for reducing hardware costs. Thus far, we have considered only wavelength-level switches. In any reasonably-sized network, the number of wavelengths on the fiber is typically much greater than the nodal degree. The result is that many wavelengths in each input fiber at any given node will be switched to the same output fiber. Rather than switching each wavelength individually, *waveband switches* can be used to switch all the wavelengths using a single port, instead of requiring a separate port for each wavelength.

Waveband switching is based on the observation that if the number of input wavelengths per fiber is large relative to the number of output fibers, many of the wavelengths will need to be switched between the same fiber pairs. Waveband switching tries to group the wavelengths into wavebands such that all wavelengths in the same waveband can be switched together, allowing the processing to be performed at this coarser waveband level and reducing the number of switches required. With waveband switching each fiber would only be demultiplexed into wavebands, and the number of switches required would equal the number of wavebands. Since the number of wavebands required is typically much smaller than the number of wavelengths, this can greatly reduce the processing and switching costs.

Here we consider the resources of interest to be the number of wavelengths and wavebands required by a given banding algorithm. Reducing the requirement for either quantity reduces the costs in the network. Conceptually, every banding algorithm can be represented by a point in a two-dimensional performance space, as illustrated in Figure 1-3, indicating the number of wavelengths and wavebands required by the algorithm. The shaded area in the figure represents the achievable region of performance over all possible algorithms. The goal is to characterize the optimal frontier of achievable performance. This frontier would give the optimal tradeoff between wavelengths and wavebands achievable.

There has been some work in the literature addressing the waveband switching problem. Many papers consider the problem of waveband allocation for static traffic. In [5, 20, 34], integer linear programming formulations are given for a variety of topologies, and the problem of optimal waveband allocation is shown to be NP-complete. In [16], efficient algorithms for dynamic traffic are considered under a simplified traffic model that limits traffic to a single source node and does not allow wavelength overprovisioning, even if it results in fewer wavebands.

In Chapter 4, we propose to consider dynamic traffic under the more general problem of determining the optimal tradeoff between wavelengths and wavebands in band switching. Furthermore, we will allow a more general traffic model where every node is permitted to send traffic into the network. We will first bound the region of achievable performance points by developing lower bounds on the minimum number

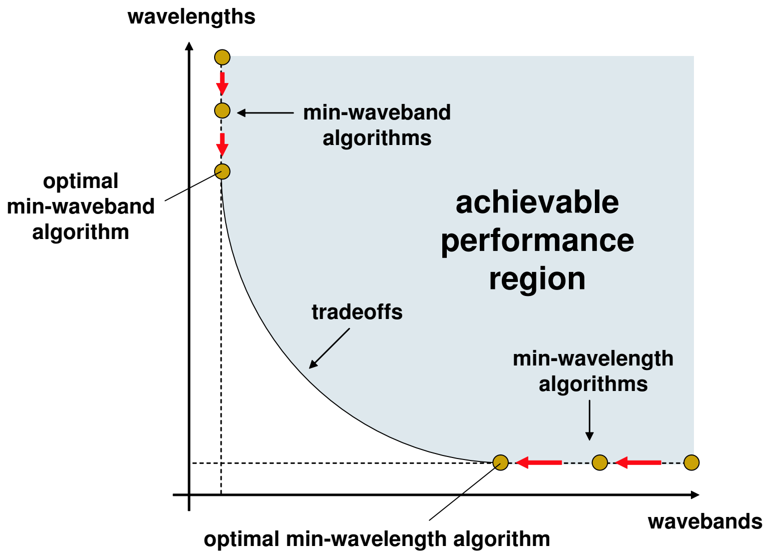


Figure 1-3: The region of achievable wavelength-waveband tradeoffs.

of wavelengths and the minimum number of wavelengths achievable in any banded system. These bounds are represented by the dotted lines in Figure 1-3.

We next consider two regimes: the minimum-wavelength regime (represented by the horizontal dotted line on the bottom), where we constrain ourselves to consider only designs for which the minimum number of wavelengths are used, and the minimum-waveband regime (represented by the vertical dotted line on the left), where we are constrained to designed for which only the minimum number of wavebands are used. We will derive the optimal algorithms in these two regimes, which will correspond to the two points (each lying on one dotted line) closest to the origin that can be practically achieved. We will also present novel algorithms which will achieve a very attractive tradeoff between these two points, demonstrating that points very close to both the minimum number of wavelengths and wavebands can be achieved.

1.4 Hybrid Static-Dynamic Provisioning

Up until this point, we have assumed that each wavelength provisioned in the network has the capability of being dynamically switched in real time. Chapter 2 presents the most relaxed conditions, where all wavelengths were individually switched. However, even in the next two chapters, each wavelength (including the bypass wavelengths of Chapter 3 and the banded wavelengths of Chapter 4) also has limited switching capability, even if it was not accessible to each node or if it could only be switched together in a band with other wavelengths.

There is an interesting alternate approach to increasing the number of calls the network can support without increasing the number of switch ports required at each node: rather than having all wavelengths be dynamically switched, a number of wavelengths could be statically provisioned. This approach is very sensible for high-traffic networks: in these cases, the traffic typically has a large mean and some variation around that mean. If the deviation from the mean is not very large, then it is reasonable to expect that a good approach would be to statically provision a number of wavelengths to support the average behavior of the traffic, and a smaller number of dynamic wavelengths to support the inevitable random variations around this average.

In general, the network can consist of a large number of nodes connected in some arbitrary fashion (see Figure 1-4). This presents a complex provisioning problem over multiple links. For simplicity, in Chapter 5 we will focus on provisioning a single shared link on a backbone network. Figure 1-5 shows a model for the shared colored link in Figure 1-4. We consider provisioning for traffic traveling from left to right along the link. Each wavelength on the link can be used to support one lightpath from one of the incoming fibers on the left side of the link to one of the outgoing fibers on the right side of the link.

Broadly speaking, wavelength provisioning can be done in one of two ways. One option is to *statically* provision a wavelength by hard-wiring the nodes at the ends of the link to always route the wavelength from a given input fiber to a given output fiber. The advantage to this is that the cost of the hardware required to support static provisioning is relatively low: no switching capability or intelligent decision-making ability is required. The downside is a lack of flexibility in using that wavelength – even if the wavelength is not needed to support a lightpath between the assigned input and output fibers, it cannot be assigned to support a lightpath between any

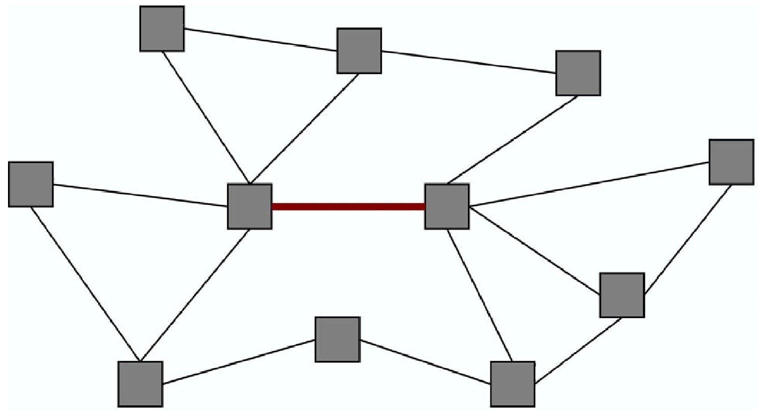


Figure 1-4: An example of a mesh optical network consisting of numerous nodes and links.

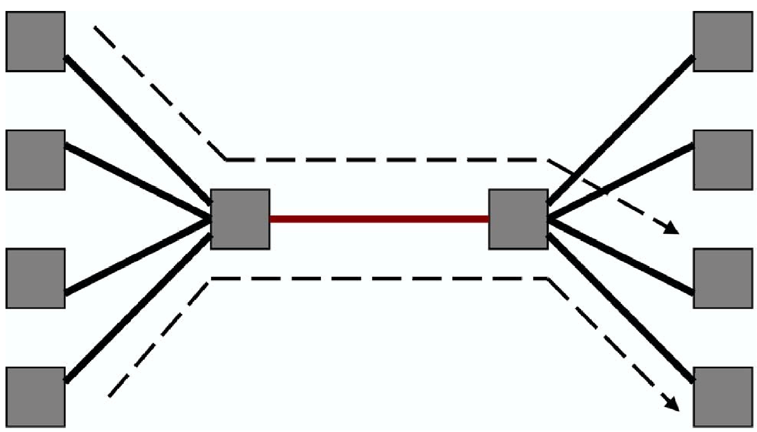


Figure 1-5: A shared-link model based on the colored link in Figure 1-4. The dotted lines denote different users of the link. Since each pair of input-output fibers comprises a different user, and there are 4 input fibers and 4 output fibers, there are a total of $4 \cdot 4 = 16$ users in this example.

other pair of fibers.

This shortcoming can be overcome by using *dynamic* provisioning. A dynamically provisioned wavelength is switched at the nodes on both sides of the link, allowing it to be dynamically assigned to support a lightpath between any source and destination fibers. Furthermore, this assignment can change over time as traffic demands change. This obviously imparts a great deal of additional flexibility. The downside is that the added switching and processing hardware makes it more expensive to dynamically provision wavelengths.

The cost of dynamic provisioning can be somewhat reduced by using waveband switching, as described in the previous section. Under such an approach, wavelengths are grouped into wavebands consisting of several wavelengths each, and all wavelengths in the same band must be switched together. This provides for limited flexibility, since a waveband can still be assigned to any user, but wavelengths within the same band cannot be shared by two different users. Since a single switch port can now accommodate all the wavelengths in a single band, the switch size (and therefore cost) can be greatly reduced.

There has been much investigation of both statically provisioned and dynamically provisioned systems in the literature [2, 4, 7, 14, 18, 21, 27, 28]. Such approaches are well-suited for cases where either the traffic is known a priori and can be statically provisioned, or is extremely unpredictable and needs to be dynamically provisioned. However, in practice, a middle ground is usually more common. Traffic reaching high-bandwidth optical networks typically consists of an amalgamation of a large number of smaller calls, and statistical multiplexing helps reduce the variance of the traffic. As a result, it is common to see traffic demands characterized by a large mean and a small variance around the mean.

A hybrid system is well suited to such a scenario. In a hybrid system, a sufficient number of wavelengths are statically provisioned to support the majority of the traffic. Then, on top of this, a smaller number of wavelengths are dynamically provisioned to support the inevitable variation in the realized traffic. Such an approach takes advantage of the relative predicability of the traffic by cheaply provisioning the majority of the wavelengths, but retains sufficient flexibility through the minority of dynamic wavelengths that significant wavelength overprovisioning is not necessary.

We begin by using an asymptotic analysis where we allow the number of users in the system become large, and consider the provisioning requirements in order for the network to be asymptotically non-blocking. We first consider the case where the shared dynamic wavelengths are individually switched, and derive expressions for the amount of static and dynamic provisioning required as a function of the traffic parameters. We then repeat for waveband-switched systems, and show that in a waveband-switched system, provisioning just a small number of switches results in performance very close to a fully switched system.

We finally consider networks with a small, finite number of users. In the finite case, strictly non-blocking performance cannot be achieved using the preceding method. However, in practical networks, we can typically tolerate a fixed probability of an overflow event (where one or more calls are blocked) occurring. We show how, given a target overflow probability, the results of the asymptotic analysis can be adapted to produce a network whose performance will meet the target requirements. We also show that by adopting this approach, a relatively small number of additional wavelengths are required (compared to the bound established by the asymptotic analysis).

Chapter 2

Routing and Wavelength Assignment

The base case under consideration is where each wavelength is fully processed at each node. In such a scenario, we can consider the figure of merit to be solely the number of wavelengths used. Under these assumptions, we have maximum flexibility in the RWA problem, and hence the performance of an optimal RWA algorithm under these conditions provides us with a lower bound on the maximum efficiency of wavelength usage. This understanding will also be useful in our subsequent consideration of reduced-complexity node architectures, since it provides a basis for comparison of wavelength efficiency.

In this chapter, we will investigate a simple yet commonly-used symmetric network architecture, the ring network. We develop theoretical bounds on the minimum number of wavelengths required to support rearrageably non-blocking traffic given full wavelength conversion, and show by construction that this bound is achievable using a minimal number of wavelength converters. Converter placement will also be investigated to determine where converters should be located within the ring, and how to modify a wavelength assignment to accommodate any external restrictions on converter placement.

2.1 System Model

We consider a bidirectional ring with N nodes. Adjacent nodes are connected by two fibers: one supporting wavelengths travelling in the clockwise direction, the other supporting wavelengths in the counterclockwise direction. The two fibers are represented by a single bidirectional link, where each link can support calls travelling in both directions on every wavelength.

A wavelength converter, if available at a given node, can be used to switch a call arriving to that node on one wavelength onto a different wavelength departing the node. If no conversion is employed, a call passing through a node on one wavelength must exit the node on the same wavelength. The cost of providing wavelength conversion from one wavelength to another is assumed to be fixed and independent of the frequency separation between the wavelengths. A traffic matrix or traffic set consists of a set of calls that need to be set up in the network. Each call consists of a source and destination pair. A traffic set is *connected* if the directed graph corresponding to the set of source-destination pairs is connected. In a *single-port* network, each node is considered to have a single tunable optical transmitter and receiver. Hence each node may at most originate one call (using any available wavelength) and receive one call (on any wavelength, possibly different from the one used by the transmitter). In a P*-port* network, each node i has P_i transmitters and receivers, and hence can transmit and receive P_i different calls. P-port networks can be either *symmetric*, where $P_i = P$ for all nodes, or *asymmetric*, where P_i can differ for each node. This is a natural problem to consider since equipment constraints limit the number of ports each node has available. The set of all traffic matrices which satisfy the P-port requirement is called the *admissible set*. Routing and assigning wavelengths to each of these traffic matrices is the RWA problem, considered in this chapter.

We consider the problem of supporting any admissible traffic set in a P-port network in a rearrangeably non-blocking fashion. In this context, there are a number of metrics which are relevant to evaluating the performance of a RWA algorithm. One is the worst-case number of wavelengths required by the algorithm – the smaller the number, the better. Another is the total number of wavelength converters the algorithm uses. Since converters are expensive, an algorithm that uses converters sparingly is preferred. Finally, in general the converter requirements may be different at each node. Certain distributions may be more desirable than others depending on the design criteria: for example, in some cases, we may want a *hub* design where all converters are placed at a single node; in others we may prefer the converters to be distributed equally at all nodes. We consider algorithms which attempt to design a RWA for these metrics.

In Section 2.2, we derive a lower bound on the number of wavelengths required to support the worst-case traffic set, and present two RWA schemes for both connected and unconnected traffic sets in single-port networks: an optimal algorithm which uses the minimum possible number of wavelengths to support all traffic sets, and a suboptimal algorithm which uses more wavelengths but requires significantly fewer converters. These results are extended to multi-port networks in Section 2.3. In Section 2.4 we develop a method for changing the location of wavelength converters in a given RWA, and apply the method to the algorithms in the previous sections.

2.2 Single-Port Ring Networks

2.2.1 The $\lceil N/4 \rceil$ Algorithm For Connected Rings

We consider here the case of a single-port network, and require that the RWA algorithm be able to route any connected traffic set in a rearrangeably non-blocking fashion. Our initial goal is to design a RWA algorithm which minimize the number of wavelengths used. The following theorem gives a lower bound on the number of wavelengths required by the worst-case traffic set for this network.

Theorem 1. *For a single-port N-node bidirectional ring, at least $\lceil N/4 \rceil$ wavelengths are required by the worst-case traffic set for N even, and $\lceil (N-1)/4 \rceil$ wavelengths for N odd.*

Proof. Consider the case where N is even, and envision a cut which divides the network into two sets of $N/2$ nodes each. Since the nodes were formed in a ring, this cut consists of two links. Consider a traffic set where each of the $N/2$ nodes in one set wishes to communicate to one of the nodes in the other set. In this case, a total of $N/2$ calls must cross the cut in either direction, for a total of N calls. Since each link in the cut can support at most two calls on a single wavelength (one clockwise, one counterclockwise), a minimum of $\lceil N/4 \rceil$ wavelengths are required to support the calls across the cut. Similar reasoning for N odd gives a bound of $\lceil (N-1)/4 \rceil$. $\square$

It is worth noting that this bound cannot be achieved by a simple routing scheme such as shortest-path. To see this, consider a ring with an even number of nodes N, and number the nodes in increasing order from 1 to N in the clockwise direction. Consider the traffic set where each node n_i sends a call to node $n_{i \oplus [(N/2)-1]}$. (We use $\oplus$ to denote addition modulo N.) Then shortest-path would route all calls in the clockwise direction, with each call requiring $(N/2)-1$ hops to accommodate it. Since there are N calls total, this would require at least $N \cdot (N/2-1)/N = (N/2)-1$ wavelengths to support it.

We next describe the operation of our first RWA algorithm and assert that it is optimal in the sense that it requires no more than the lower bound of $\lceil N/4 \rceil$ wavelengths. The proof follows the description.

Consider an arbitrary connected traffic set $\{c_1, c_2, \ldots, c_N\}$ consisting of source-destination pairs c_i. We term a pair of calls *adjacent* if the destination node of the first call is the source node of the second. In a connected traffic set, it is always possible to traverse all calls in the traffic set in adjacent order; i.e., there are no sub-cycles within the traffic set. Therefore without loss of generality we can renumber the calls so that they are indexed in adjacent order; that is, c_i is adjacent to $c_{i \oplus 1}$ for every i.

Denote the number of hops required to route a particular call c_i in the clockwise direction by L_i. Denote the average number of hops required in the clockwise direction by

$$\bar{L} = \frac{\sum_{i=1}^{N} L_i}{N}$$

Then the algorithm is as follows:

THE $\lceil N/4 \rceil$ ALGORITHM

1. TRAFFIC SET PARTITIONING: Let $k = \min\{\lfloor N^2/4\bar{L} \rfloor, N\}$. Find a set of k adjacent calls with average clockwise hop length $\tilde{L}$ less than or equal to $\bar{L}$. Call this set the *clockwise set*. Designate all calls not contained in the clockwise set to be members of the *counterclockwise set*. (We will shortly show that such sets always exist.)

2. ROUTING: Route all calls in the clockwise set in the clockwise direction. Route all calls in the counterclockwise set in the counterclockwise direction.

3. WAVELENGTH ASSIGNMENT (CLOCKWISE SET): Assign wavelengths to calls using a *forward pass* and a *reverse pass* as follows: Index all calls c_m in the clockwise set in adjacent order. Index the wavelengths λ_n in arbitrary order. Initialize $i = 1$ and $j = 1$.

   (a) FORWARD PASS: In this part, beginning with the first call and proceeding in adjacent order, assign as many calls as possible to the first wavelength without using conversion. When a call cannot be fully assigned to the wavelength, assign it entirely to the next wavelength (without conversion) and repeat, until all $\lceil N/4 \rceil$ wavelengths are used. This is made explicit below:

      i. Assign call c_i entirely to λ_j without using any conversion.

      ii. Increment i: $i \leftarrow i + 1$.

      iii. If call c_i can be assigned entirely to λ_j without conversion, goto (i). Otherwise continue.

      iv. Increment j: $j \leftarrow j + 1$.

      v. If $j \leq \lceil N/4 \rceil$, goto (i). Otherwise stop.

(b) REVERSE PASS: In this part, the remaining calls are assigned to the wavelengths in the reverse of the order they were filled in the forward pass, using converters as necessary. More explicitly:

   i. Assign as much of the unassigned portion of call c_i to λ_j as possible.
   ii. If c_i is completely assigned, increment i and goto (i). Otherwise continue.
   iii. Using a wavelength converter, convert the last hop of c_i allocated in (i) from λ_j to λ_{j-1}.
   iv. Decrement j: $j \leftarrow j - 1$.
   v. If all calls have been assigned, stop. Otherwise goto (i).

4. WAVELENGTH ASSIGNMENT (COUNTERCLOCKWISE SET): Repeat Step 3 with the counterclockwise set in the counterclockwise direction.

We will refer to this as the $\lceil N/4 \rceil$ algorithm. The following example illustrates the use of the $\lceil N/4 \rceil$ algorithm for a particular traffic set.

Example 1. *Consider an 8-node ring, where* $\lceil N/4 \rceil = 2$. *Number the nodes from 1 to 8 in the clockwise direction. Consider a traffic set consisting of the following calls, listed in adjacent order: (1,4), (4,6), (6,2), (2,5), (5,8), (8,3), (3,7), and (7,1). We will apply the* $\lceil N/4 \rceil$ *algorithm to this problem.*

The average clockwise hop length $\bar{L} = 3$, *and* $k = \min\{\lfloor N^2/4\bar{L} \rfloor, N\} = \min\{\lfloor 16/3 \rfloor, 8\} = \min\{5, 8\} = 5$. *Choose the clockwise set to be the set of calls* $\{$*(1,4), (4,6), (6,2), (2,5), (5,8)*$\}$, *with average hop length* $\tilde{L} = (3+2+4+3+3)/5 = 3 \leq \bar{L}$. *The counterclockwise set then consists of the remaining calls,* $\{$*(8,3), (3,7), (7,1)*$\}$. *Note that the average hop length obeys* $\hat{L} = (3+4+2)/3 \geq \bar{L}$ *in the clockwise direction.*

In the forward pass on the clockwise set, calls (1,4) and (4,6) are assigned to the first wavelength, while (6,2) and (2,5) are assigned to the second wavelength. This situation is shown in Figure 2-1(a). In the reverse pass, the final call (5,8) is assigned partly on each wavelength and employs a converter at node 6. The final RWA for the clockwise set is shown in Figure 2-1(b).

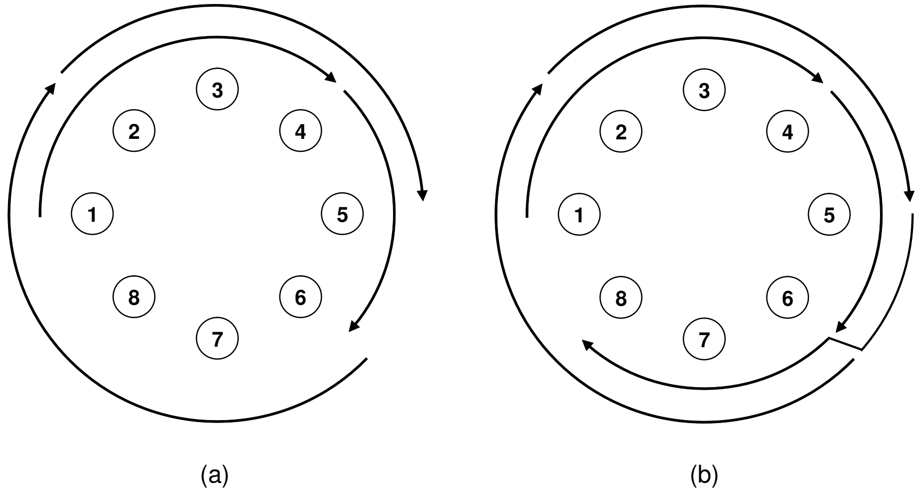


Figure 2-1: (a) The routing and wavelength assignment of calls in the clockwise set after the forward pass. The inner arrows represent calls on λ_1, the outer arrows are calls on λ_2. (b) The complete RWA on the clockwise direction after the backward pass.

In the forward pass on the counterclockwise set, calls (8,3) and (3,7) are assigned to the first and second wavelengths, respectively. In the reverse pass, (7,1) is assigned partly to both and again requires a converter.

We make two claims regarding this algorithm. First, it is always possible to find a set of $k = \min\{\lfloor \frac{N^2}{4\bar{L}} \rfloor, N\}$ adjacent calls with average clockwise hop length less than or equal to $\bar{L}$. Second, using this algorithm, any admissible traffic set requires at most $\lceil N/4 \rceil$ wavelengths and $\lceil N/2 \rceil - 2$ converters. These claims will be formalized as Lemma 9 and Theorem 2.

Lemma 1. *There exists a set of n adjacent calls with average clockwise hop length $\tilde{L}$ less than or equal to the average clockwise hop length of the entire traffic set $\bar{L}$, for any $0 \leq n \leq N$. Furthermore, the $N - n$ calls in the complement of that set have average clockwise hop length $\hat{L} \geq \bar{L}$.*

Proof. We will conduct a proof by contradiction. Suppose there did not exist any set of n adjacent pairs with average hop length less than $\bar{L}$. In particular, this would imply that

$$\begin{aligned}
\frac{1}{n} \cdot (L_1 + ... + L_n) &> \bar{L} \\
\frac{1}{n} \cdot (L_2 + ... + L_{n+1}) &> \bar{L} \\
&\cdots \\
\frac{1}{n} \cdot (L_{N-n+2} + ... + L_N + L_1) &> \bar{L} \\
&\cdots \\
\frac{1}{n} \cdot (L_N + L_1 + ... + L_{n-1}) &> \bar{L}
\end{aligned}$$

Summing the entire set of N inequalities, we obtain

$$L_1 + \cdots + L_N > \bar{L}N$$

where the coefficient of each term L_i is unity, since each L_i is involved in exactly n of the inequalities and is scaled by a factor of $\frac{1}{n}$. Equivalently,

$$\frac{1}{N} \cdot (L_1 + \cdots + L_N) > \bar{L}$$

But since by definition $\bar{L}$ is the average hop length, this cannot be true. Hence there must exist a set of n adjacent pairs with average hop length less than $\bar{L}$.

The second half of the proof also uses contradiction. Suppose for the remaining $N - n$ calls, the average clockwise hop length $\hat{L} < \bar{L}$. From the definitions of $\hat{L}$ and $\tilde{L}$, we have that

$$\begin{aligned}
\hat{L} = L_{n+1} + \cdots + L_N &< (N-n)\bar{L} \\
\tilde{L} = L_1 + \cdots + L_n &\leq n\bar{L}
\end{aligned}$$

Combining the preceding two inequalities and dividing by N, we then obtain

$$\frac{1}{N} \cdot (L_1 + \cdots + L_N) < \bar{L}$$

which contradicts the definition of $\bar{L}$ being the average hop length. $\square$

For our purposes, we will mainly be interested in applying Lemma 9 for the case of $n = k$ in the proof of the following theorem.

Theorem 2. *Given any connected traffic set, the* $\lceil N/4 \rceil$ *algorithm requires only* $\lceil N/4 \rceil$ *wavelengths and at most* $\lceil N/2 \rceil - 2$ *converters.*

Proof. By Lemma 9, it is always possible for the algorithm to find valid clockwise and counterclockwise sets. Consider first the clockwise set. For simplicity, consider those cases where the total number of wavelengths $N/4$ is an integer. (For all other cases, fictitious nodes can be added to increase $N/4$ to the nearest integer.) First note that $N/4$ wavelengths in an N-hop ring can support $N^2/4$ contiguous hops of traffic. By choice of the clockwise set, the average clockwise hop length in the clockwise direction $\tilde{L} \leq \bar{L}$. Then the total number of hops required to accommodate the clockwise set, denoted by D_C, is

$$\begin{aligned} D_C &= k\tilde{L} \\ &\leq k\bar{L} \\ &\leq \left\lfloor \frac{N^2}{4\bar{L}} \right\rfloor \cdot \bar{L} \\ &\leq \frac{N^2}{4} \end{aligned}$$

Since all required hops are contiguous due to the adjacency of all calls in the set, the clockwise set fits in $N/4$ wavelengths.

Next consider the counterclockwise set, which contains the remaining $N - k$ calls. If $k = N$, then $N - k = 0$ and the counterclockwise set is empty and requires no wavelengths, completing the proof. Therefore assume $k = \lfloor N^2/4L \rfloor$. Denote the average clockwise hop length $\hat{L}$; this implies that the average counterclockwise hop length is $N - \hat{L}$. Since by Lemma 9 $\hat{L} \geq \bar{L}$, it must be that the average counterclockwise hop length $N - \hat{L} \leq N - \bar{L}$. Denote the total number of contiguous hops required to accommodate the counterclockwise set by D_W. Then,

$$
\begin{aligned}
D_W &= (N-k)\cdot(N-\hat{L}) \\
&\leq (N-k)\cdot(N-\bar{L}) \\
&= \left(N - \left\lfloor \frac{N^2}{4\bar{L}} \right\rfloor\right)\cdot(N-\bar{L})
\end{aligned}
$$

We show in Section 2.6 that for N even, the last quantity is maximized at $\bar{L} = N/2$, giving us

$$
D_W \leq \left(N - \left\lfloor \frac{N^2}{4\bar{L}} \right\rfloor\right)\cdot(N-\bar{L}) \leq \frac{N^2}{4}
$$

which also fits in the $\lceil N/4 \rceil$ wavelengths. Note that there is no loss of generality in the assumption of N even, as explained earlier and in Section 2.6.

By construction, the $\lceil N/4 \rceil$ algorithm requires up to one converter on each wavelength (except the last) in each direction, for a total of $2\lceil N/4 \rceil - 2$ converters. Additionally, consider the location of the converters: each converter, where needed, is located at the destination node of the last call on each wavelength after the forward pass on the clockwise and counterclockwise sets. Since we are dealing with a single-port network, each node is the destination of no more than a single call. This implies that no node requires more than a single converter at most. □

Later, in Section 2.4, we will show how the wavelength assignment can be modified to distribute the $2\lceil N/4 \rceil - 2$ converters almost arbitrarily among all nodes in the ring.

2.2.2 The $2\lceil N/7 \rceil$ Algorithm For Connected Rings

Although the $\lceil N/4 \rceil$ algorithm achieves the minimum number of wavelengths, it may require as many as $2\lceil N/4 \rceil - 2$ converters to do so. Since converters may be costly, it is desirable to reduce the number of converters required. In [24] an algorithm is provided that does not require converters but uses $\lceil N/3 \rceil$ wavelengths. Motivated by a desire to find a compromise between these two extremes, we present our next algorithm that requires $2\lceil N/7 \rceil$ wavelengths and only $\lceil N/7 \rceil$ converters.

We will begin by restating a result from [24] regarding the routing of adjacent pairs and giving a new lemma on routing adjacent triplets. Then, using these results, we will give an algorithm which divides the connected traffic set into smaller sets of 7 adjacent calls and routes each set of 7 calls onto two wavelengths (in each direction).

Lemma 2. *Given an adjacent pair of calls, it is possible to fit the calls onto a single wavelength in either the clockwise or counterclockwise direction with no wavelength conversion.*

Proof. See [24]. □

Lemma 3. *Given a direction around the ring and given an adjacent triplet of calls, if it is not possible to fit the calls into a single wavelength (using no converters) in that direction, then it is possible to fit the calls into two wavelengths (using a single converter) in the opposite direction.*

Proof. Denote the calls by their source-destination pairs as follows: (n_1, n_2), (n_2, n_3), and (n_3, n_4). Without loss of generality, suppose by Lemma 2 that (n_1, n_2) and (n_2, n_3) fit on a single wavelength in the clockwise direction. (If the opposite is true, then simply reverse the clockwise/counterclockwise directions to follow.) We prove the lemma first for the choice of the clockwise direction, then the counterclockwise.

CLOCKWISE: Suppose the choice of direction was clockwise. If all three calls can be routed in the clockwise direction, then this part of the proof is complete. Suppose they cannot; i.e., part of the path (n_3, n_4) overlaps part of the path (n_1, n_2) in the clockwise direction. This implies that, travelling in a clockwise direction from node n_3, we first encounter node n_1 before node n_4. Reversing the directions, it must therefore also be the case that travelling in a counterclockwise direction from n_3, we first encounter node n_4 before node n_1. This is illustrated in Figure 2-2.

We can route (n_1, n_2) and (n_2, n_3) each onto separate wavelengths λ_1 and λ_2 in the counterclockwise direction. This leaves the links between n_2 to n_1 on λ_1 and n_3

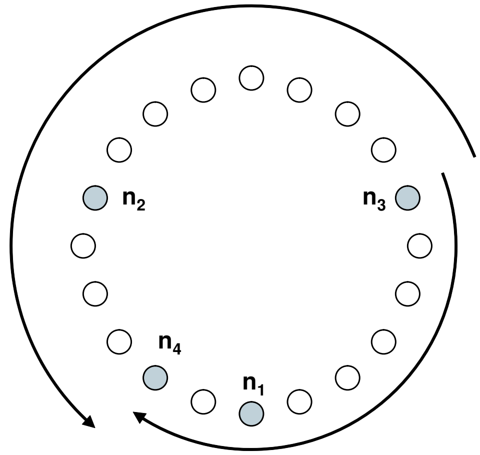


Figure 2-2: Beginning at node n_3, since we first encounter node n_1 before n_4 when travelling in the clockwise direction, we must encounter n_4 before n_1 when travelling in the counterclockwise direction.

to n_2 free on λ_2. Since travelling in the counterclockwise direction we reach node n_4 before n_1, the third call (n_3, n_4) can fit into the free links on λ_1 and λ_2 in the counterclockwise direction using a converter at node n_2.

COUNTERCLOCKWISE: Next consider if the choice was counterclockwise. It is not possible to fit all calls into a single wavelength in this direction, so therefore we must show it is possible to fit all calls in two wavelengths in the clockwise direction. This is done by noting that since by assumption the first two calls can fit on a single wavelength in the clockwise direction, the third can fit alone on a second wavelength. □

Figures 2-3 and 2-4 illustrate examples of applying Lemmas 2 and 3, respectively. We will now use the two preceding lemmas to describe a method for fitting any set of 7 adjacent calls onto at most two wavelengths.

Theorem 3. *Given a set of 7 adjacent calls, the entire set can be routed using at most two wavelengths (in each direction).*

Proof. We will provide a proof by construction. Consider the first four adjacent calls. Divide them into two adjacent pairs. By Lemma 2, each pair can be routed using a single wavelength in either the clockwise or counterclockwise direction. First suppose

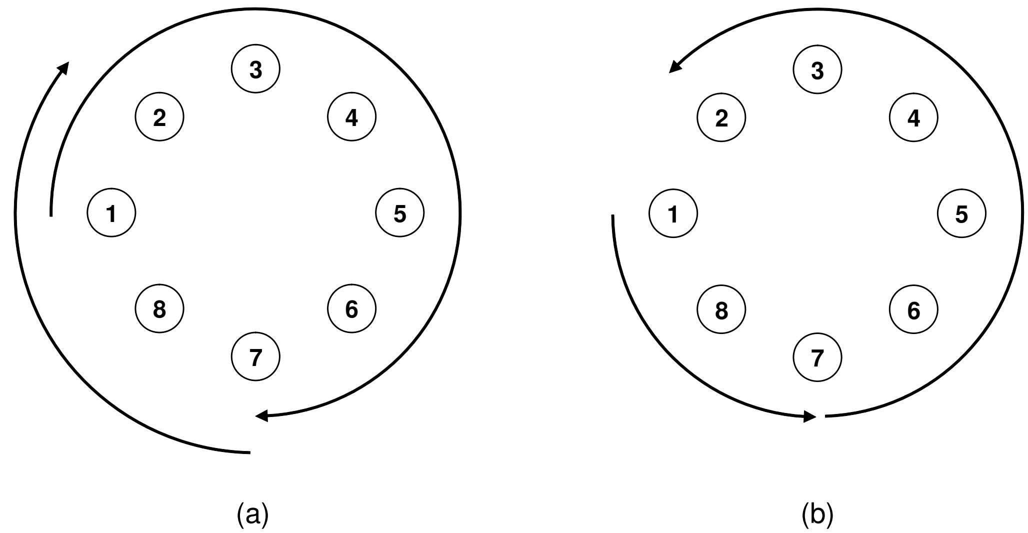


Figure 2-3: (a) This adjacent pair cannot be placed on a single wavelength in the clockwise direction. (b) Therefore by Lemma 2, it can fit without converters on a single wavelength in the counterclockwise direction.

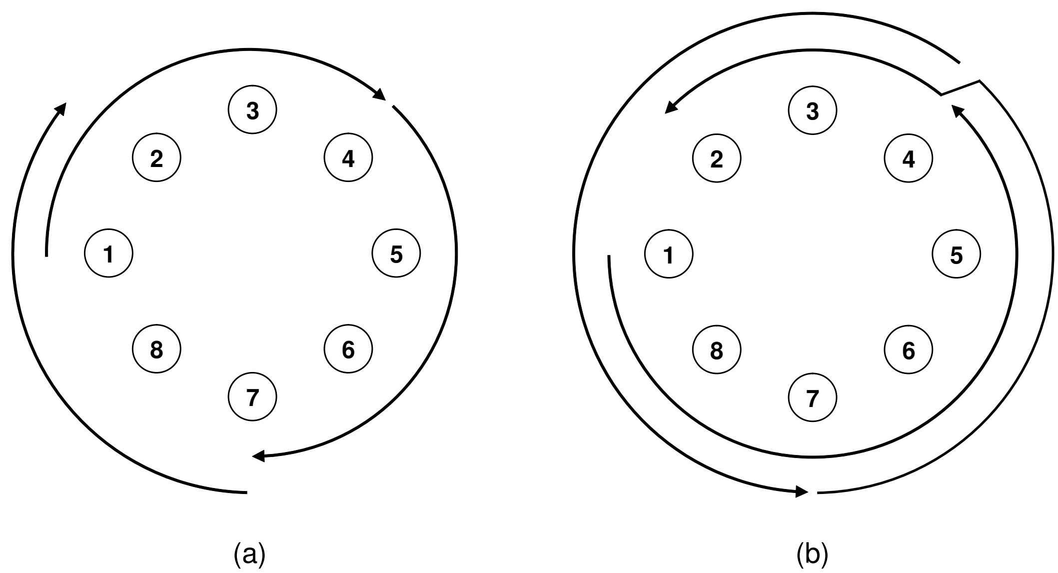


Figure 2-4: (a) The adjacent triplet $(n_1, n_4), (n_4, n_7), (n_7, n_2)$ cannot be placed on a single wavelength in the clockwise direction. (b) Therefore by Lemma 3, it can fit on two wavelengths in the counterclockwise direction using only a single converter. The converter is required at node 4 in this case. Notice also that the triplet can fit using two wavelengths in the clockwise direction.

that the two wavelengths are in different directions. Then they can share the same wavelength, and the first four paths can be routed using a single wavelength. Of the remaining three calls, by Lemma 2 the first adjacent pair can again be fit on a single wavelength in one direction; placing the remaining call on the same wavelength in the opposite direction completes the construction in this case.

Next suppose that the first two pairs can only fit on single wavelengths in the same direction. Without loss of generality, let this direction be clockwise. Consider the remaining adjacent triplet.

If these calls can be placed onto a single wavelength in the clockwise direction, then do so. Also place the first pair on a second wavelength in the clockwise direction. Then place the two calls in the second pair on the same two wavelengths in the counterclockwise direction, each using their own wavelength.

If the last three calls cannot be placed onto a single wavelength in the clockwise direction, then by Lemma 3 they can be placed onto at most two wavelengths in the counterclockwise direction. The first two pairs can then be routed onto the same two wavelengths in the clockwise direction, each pair using its own wavelength. □

In general, we can route any connected traffic set by dividing it into adjacent sets of 7 calls and applying the construction in the proof of Theorem 3 to each set. We will call this the $2\lceil N/7 \rceil$ algorithm.

THE $2\lceil N/7 \rceil$ ALGORITHM

1. Divide the traffic set into $c = \lceil N/7 \rceil$ adjacent sets of 7, each denoted by C_j, $1 \leq j \leq c$. Let $i = 1$.

2. Route each set of 7 calls using 2 wavelengths, following the proof of Theorem 3, for a total of $2\lceil N/7 \rceil$ wavelengths.

Converter Requirements: During the RWA construction, the traffic set is divided into $\lceil N/7 \rceil$ sets of 7 adjacent calls; each set of 7 calls uses at most a single converter. Using these facts, we can show that the total number of converters required is upper-bounded by $\lfloor N/7 \rfloor$.

To see why we can use only $\lfloor N/7 \rfloor$ rather than $\lceil N/7 \rceil$ converters, we need to consider two cases: where N is and is not divisible by 7. Supposing N is divisible by 7, $\lfloor N/7 \rfloor = \lceil N/7 \rceil$ and the distinction is irrelevant. Next suppose N is not divisible by 7. Then the first $\lceil N/7 \rceil - 1$ sets require at most $\lceil N/7 \rceil - 1 = \lfloor N/7 \rfloor$ converters. The last set has at most 6 adjacent calls. (If it has less, insert fictitious calls.) Further divide this set into two sets of 3 adjacent calls. Each set of 3 calls can be routed using a single wavelength without conversion by putting the first two adjacent calls onto a single wavelength in one direction without conversion (guaranteed by Lemma 2) and putting the remaining call in the other direction on the same wavelength.

The converter in each set, if required, is located at the destination of one of the calls. Since we are considering a single-port network wherein each node form the destination of only one call in the traffic set, no node requires more than one converter. We later show in Section 2.4 how the wavelength assignment can be modified to distribute the $\lfloor N/7 \rfloor$ converters almost arbitrarily among all nodes.

2.2.3 Handling Unconnected Traffic Sets

Thus far we have limited our discussion to connected traffic sets. We next consider unconnected traffic sets; that is, traffic sets where in the corresponding directed graph there exist nodes which do not communicate. For single-port traffic, we will see that this implies that the traffic set is composed of a number of cycles.

We consider only *maximal* traffic sets; i.e., traffic sets containing the maximum number of calls given the single-port restriction. Note that any non-maximal traffic set can be converted to a maximal set by adding fictitious calls; hence it is sufficient to consider the RWA of maximal sets. We can construct the cycles as follows:

1. Initialize $i = 1$.

2. Choose an arbitrary node, called the cycle start node. Find the call originating at that node. Move to the destination of that call. Now find the call originating at this new node, and move to the destination of that call. Repeat. By the maximal assumption, each node must originate a call, so this is always possible.

The cycle is complete when the start node is revisited. Designate all calls traversed in this step as members of the cycle C_i.

3. Remove all calls in C_i from the traffic set. By the single-port assumption, since each node encountered in the previous step is the source and destination of some call in C_i, they are not involved in any remaining calls in the traffic set.

4. If the traffic set is not yet empty, increment $i \leftarrow i + 1$ and goto Step 2.

This construction divides the traffic set into cycles involving disjoint sets of nodes. Next we will give a method for dealing with traffic sets with cycles by using an additional wavelength to turn it into a different RWA problem for a connected traffic set that does not contain cycles. The connected traffic set can then be processed using either of the previous algorithms.

Theorem 4. *Suppose there exists an algorithm that uses at most W wavelengths for any admissible connected traffic set in a single-port ring network. Then any admissible traffic set with cycles can be routed using at most $W + 1$ wavelengths with the addition of a number of converters equal to the number of cycles.*

Proof. The proof is by construction using the following algorithm.

Step 1 - CYCLE FORMATION: Consider a traffic set with c cycles. Group the calls into sets based on which cycle they belong to. Number these cycles C_1, C_2, ... , C_c. From each set, arbitrarily choose a single call and denote the source and destination nodes of that call by s_i and d_i, respectively, for the set i. Without loss of generality, renumber the cycles so that d_1, ... , d_c are in *counterclockwise order*; i.e. after renumbering, travelling counterclockwise around the ring beginning with d_1, one encounters each d_i in order of increasing index i.

Step 2 - SUPERCYCLE FORMATION: The idea is that we will break each cycle at the call (s_i,d_i) and connect it to the next cycle, thus forming a single connected *supercycle*. Consider a given cycle C_i. Remove the call (s_i,d_i) from the traffic set, and replace it with a new call $(s_i, d_{i\oplus 1})$. This connects all nodes in cycle C_i with cycle $C_{i\oplus 1}$. Repeat for all cycles. At the end of this procedure, we have formed a new traffic set called the supercycle, denoted by C_S. Note that the supercycle is also a maximal, admissible traffic set that obeys the single-port restrictions, since essentially all it did was permute the destinations of the various (s_i,d_i) calls of the original set.

Step 3 - RESIDUAL SET: We next need to add a set of additional calls, which we call the *residual set* C_R, to make $C_S \cup C_R$ equivalent to the original traffic set. The residual set consists of calls $(d_{i\oplus 1}, d_i)$ for $1 \leq i \leq c$. Then for a given cycle C_i, we can combine the calls $(s_i, d_{i\oplus 1})$ and $(d_{i\oplus 1}, d_i)$ from C_S and C_R, respectively, to form the original call (s_i, d_i). At most a single converter is needed at $d_{i\oplus 1}$ if the two calls are on different wavelengths.

Step 4 - RWA OF C_S AND C_R: The RWA algorithm for connected traffic sets can be used on C_S using at most W wavelengths by assumption. Thus it remains only to show that C_R can be fit onto a single additional wavelength. The calls in this set consist of (d_c, d_{c-1}), (d_{c-1}, d_{c-2}), $\ldots$, (d_3, d_2), (d_2, d_1). Note that this traffic set simply traverses all the d_i's in descending order. Since the d_i's were chosen in counterclockwise order by ascending i, it follows that they must be in clockwise order by descending i. Therefore all calls in T_1 can be fit onto a single wavelength in the clockwise direction. □

Corollary 1. *The* $\lceil N/4 \rceil$ *algorithm can handle unconnected traffic sets using at most* $\lceil N/4 \rceil + 1$ *wavelengths.*

Corollary 2. *The* $2\lceil N/7 \rceil$ *algorithm can handle unconnected traffic sets using at most* $\lceil 2N/7 \rceil + 1$ *wavelengths.*

The following example demonstrates the application of this approach to a traffic set with two cycles.

Example 2. *Consider an 8-node ring with nodes numbered from 1 to 8 in the clockwise direction. Consider a traffic set consisting of the following calls, listed in adjacent order: (1,4), (4,6), (6,2), (2,5), (5,1), (8,3), (3,7), and (7,8). Note that the traffic set has two cycles:* C_1 = {*(1,4), (4,6), (6,2), (2,5), (5,1)*}, *and* C_2 = {*(8,3), (3,7), (7,8)*}. *We arbitrarily choose the calls (1,4) and (8,3) from* C_1 *and* C_2, *respectively.*

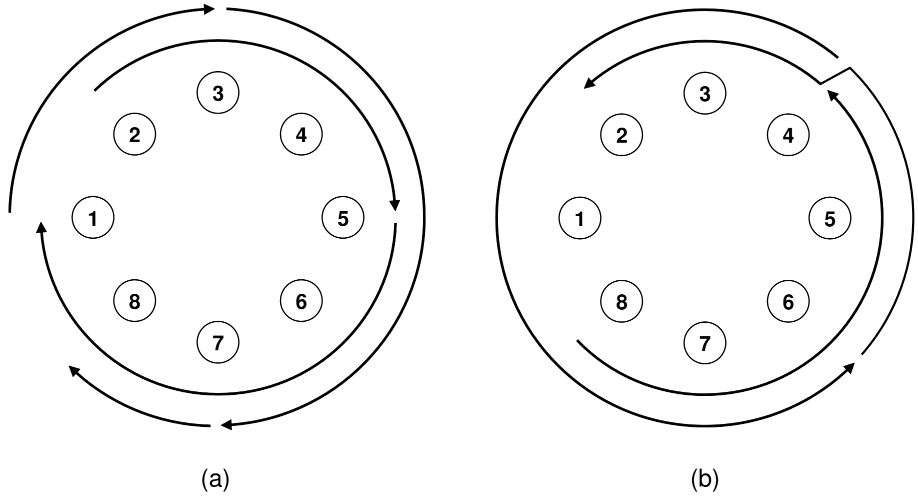


Figure 2-5: The RWA for superset T_S of Example 2.

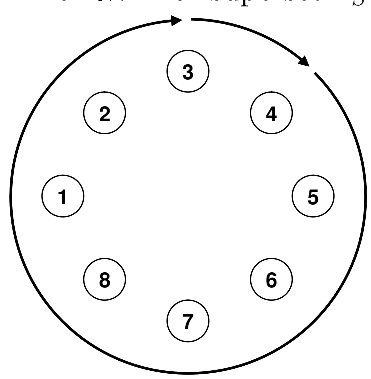


Figure 2-6: The RWA of residual set T_R.

Then $d_1 = 4$, and $d_2 = 3$. Since there are only two nodes, they are trivially in counterclockwise order and we do not need to renumber the cycles.

In addition to the previously noted values of d_1 and d_2, we also have that $s_1 = 1$ and $s_2 = 8$. Following the preceding approach, in the superset call (1,4) becomes $(s_1, d_2) = (1,3)$. Similarly, (8,3) becomes $(s_2, d_1) = (8,4)$. The superset is $T_S = \{(1,3), (4,6), (6,2), (2,5), (5,1), (8,4), (3,7), (7,8)\}$. Reordered into adjacent order, we have $T_S = \{(8,4), (4,6), (6,2), (2,5), (5,1), (1,3), (3,7), (7,8)\}$.

The residual set is $T_R = \{(d_1, d_2), (d_2, d_1)\} = \{(3,4), (4,3)\}$.

We can now route T_S using any algorithm we choose. Here we will route it using the $\lceil N/4 \rceil$ algorithm. The set T_R can by choice fit into a single wavelength. The RWA

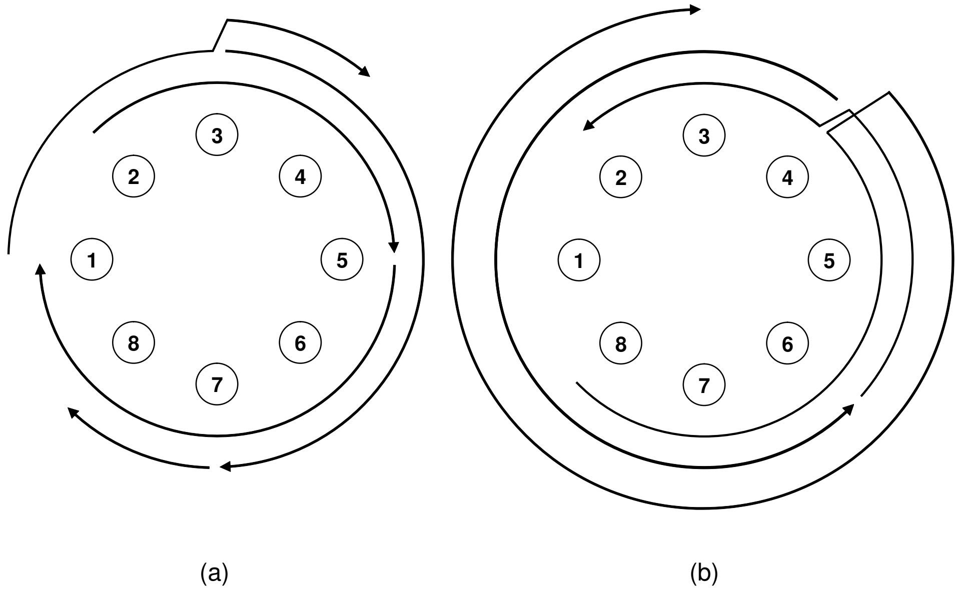


Figure 2-7: (a) and (b) show the final RWA for Example 2 in the clockwise and counterclockwise directions, respectively. Note that although the call (8,3) in (b) ended up being routed partly in the counterclockwise direction and partly in the clockwise direction, the hops in the clockwise direction do not require an additional wavelength since those hops are free on one of the existing wavelengths in (a). Also note that the RWA could be simplified by routing call (8,3) entirely in the clockwise direction, although this does not result in a savings in total wavelengths used.

for T_S and T_R are illustrated in Figures 2-5 and 2-6 respectively. Finally, the calls that were split during the creation of T_S and T_R are reconnected using wavelength converters in Figure 2-7.

Converter Requirements: By construction, one converter is required per cycle in addition to any converter requirements by the RWA algorithm.

2.3 Multi-Port Ring Networks

2.3.1 The $\lceil PN/4 \rceil$ Algorithm

1) Symmetric Multi-Port Networks: We first consider the case of connected symmetric P-port networks. By symmetric, we mean that each node has the same number of ports P. In such a network, each node has P transmitters and receivers, and can therefore send and receive P calls. Since each node is the source of at most P calls, and there are N nodes, a full traffic set contains at most PN calls. Again using a cut-set bound, it is apparent that a minimum of $\lceil PN/4 \rceil$ wavelengths are required to support the worst-case traffic set.

If the logical topology is connected, then the directed graph contains a directed Euler trail [37] which contains all edges of the graph. By finding and following the Euler trail, we can obtain the PN calls in adjacent order. We can apply a modified version of the $\lceil N/4 \rceil$ algorithm, which we will call the $\lceil PN/4 \rceil$ algorithm, to this traffic set.

THE $\lceil PN/4 \rceil$ ALGORITHM

1. TRAFFIC SET PARTITIONING: Let $k = \min\{\lfloor PN^2/4\bar{L} \rfloor, PN\}$. Find a set of k adjacent calls with average clockwise hop length $\tilde{L}$ less than or equal to $\bar{L}$. Call this set the *clockwise set*. Designate all calls not contained in the clockwise set to be members of the *counterclockwise set*.

2. ROUTING: Route all calls in the clockwise set in the clockwise direction. Route all calls in the counterclockwise set in the counterclockwise direction.

3. WAVELENGTH ASSIGNMENT: Assign wavelengths to calls using a forward and reverse pass on both the clockwise and counterclockwise sets, as in the original $\lceil N/4 \rceil$ algorithm.

This algorithm requires at most $\lceil PN/4 \rceil$ wavelengths. The proof follows the same procedure as Section 2.2.1.

For the $\lceil PN/4 \rceil$ algorithm, up to one converter on each wavelength (except the last) is required in each direction, for a total of $2\lceil PN/4 \rceil - 2$ converters. However, since we have a P-port network, similar examination of the construction of the wavelength assignment shows that since each node can be the destination of up to P calls, it may require at most P converters. Again, in Section 2.4 we will show how the wavelength assignment can be modified to distribute the $2\lceil PN/4 \rceil - 2$ converters nearly arbitrarily among all nodes. In particular, a modified wavelength assignment can be given that requires no more than $\min\{\lceil P/2 \rceil + 1, P\}$ converters per node.

The $\lceil PN/4 \rceil$ algorithm can also be applied to general unconnected networks containing cycles by using the approach of Section 2.2.3, where one additional wavelength is used to convert the traffic set into a connected traffic set.

2) General Multi-Port Networks: We next consider general networks where each node i has P_i ports, and is able to transmit and receive at most P_i calls. Under this model, the nodes can now be heterogeneous, and consequently it allows the model a great deal of generality.

Let $P_{tot} = \sum_{i=1}^{N} P_i$ be the total number of calls in the system. The following theorem states that for any admissible traffic set, connected or unconnected, it is possible to obtain a RWA for any admissible traffic set using at most $\lceil P_{tot}/4 \rceil + 1$ wavelengths.

Theorem 5. *For a general multi-port network with a traffic set containing a maximum of P_{tot} calls, the $\lceil PN/4 \rceil$ algorithm requires at most $\lceil P_{tot}/4 \rceil + 1$ wavelengths to provide a RWA for any arbitrary admissible traffic set.*

Proof. First, if the traffic set is unconnected, we use an approach similar to the one in Section 2.2.3 to turn it into a connected set. This requires using a single additional wavelength in the clockwise direction.

From this point on, we can assume that the traffic set is connected, and apply the $\lceil PN/4 \rceil$ algorithm, with the only difference being that the clockwise set is chosen to be of size $k = \min\{\lfloor P_{tot}N/4\bar{L} \rfloor, P_{tot}\}$ calls. By a proof similar to the one used for Lemma 9, it can be shown that the existence of a clockwise and counterclockwise set is guaranteed. Thus it remains only to show that no more than $\lceil P_{tot}/4 \rceil + 1$ wavelengths are required by both the clockwise and counterclockwise sets.

First consider the clockwise set. Since the total number of calls is P_{tot}, and the average (clockwise) hop length is at most $\bar{L}$, then the number of contiguous clockwise hops required is

$$
\begin{aligned}
D_C &\leq \lfloor P_{tot} N/4\bar{L} \rfloor \cdot \bar{L} \\
&\leq (P_{tot} N/4\bar{L}) \cdot \bar{L} \\
&= \frac{P_{tot} N}{4}
\end{aligned}
$$

Since each wavelength can support N contiguous hops of traffic, no more than $\lceil (P_{tot} N/4)/N \rceil = \lceil P_{tot}/4 \rceil$ wavelengths are required in the clockwise direction.

Next consider the counterclockwise direction. Again if $k = P_{tot}$ the counterclockwise set is empty, so the only case of interest is when $k = \lfloor P_{tot} N/4\bar{L} \rfloor$. Here the total number of calls is $P_{tot} - \lfloor P_{tot} N/4\bar{L} \rfloor$, and the average (counterclockwise) hop length is at most $\bar{L}$, so the number of contiguous counterclockwise hops required is

$$
D_W \leq \left(P_{tot} - \left\lfloor \frac{P_{tot} N}{4\bar{L}} \right\rfloor \right) \cdot (N - L)
$$

Applying the inequality $\lfloor x \rfloor > x - 1$ and proceeding,

$$
\begin{aligned}
D_W &< \left(P_{tot} - \frac{P_{tot} N}{4\bar{L}} + 1 \right) \cdot (N - \bar{L}) \\
&= \left(P_{tot} - \frac{P_{tot} N}{4\bar{L}} \right) \cdot (N - \bar{L}) + (N - \bar{L}) \\
&< \left(P_{tot} - \frac{P_{tot} N}{4\bar{L}} \right) \cdot (N - \bar{L}) + N
\end{aligned}
$$

where in the last line we used the fact that $\bar{L} > 0$. Next, to eliminate the dependence on $\bar{L}$, we would like to maximize the right-hand side over $\bar{L}$. To do this, we take the derivative with respect to $\bar{L}$ and set it to zero:

$$
\begin{aligned}
\left(\frac{P_{tot}N}{4\bar{L}^2}\right)(N-\bar{L}) - \left(P_{tot} - \frac{P_{tot}N}{4\bar{L}}\right) &= 0 \\
\frac{P_{tot}N^2}{4\bar{L}^2} - \frac{P_{tot}N}{4\bar{L}} - P_{tot} + \frac{P_{tot}N}{4\bar{L}} &= 0 \\
\frac{P_{tot}N^2}{4\bar{L}^2} - P_{tot} &= 0 \\
\bar{L}^2 &= \frac{N^2}{4} \\
\Rightarrow \bar{L} &= \frac{N}{2}
\end{aligned}
$$

Knowing that the maximizing value of $\bar{L}$ is $N/2$, we substitute that value back into the original equation to obtain

$$
\begin{aligned}
D_W &< \left(P_{tot} - \frac{P_{tot}N}{4(N/2)}\right) \cdot (N - (N/2)) + N \\
&= \left(P_{tot} - \frac{P_{tot}}{2}\right)\left(\frac{N}{2}\right) + N \\
&= \frac{P_{tot}N}{4} + N
\end{aligned}
$$

The total number of required wavelengths is then

$$
\begin{aligned}
\left\lceil \frac{D_W}{N} \right\rceil &= \left\lceil \frac{P_{tot}}{4} + 1 \right\rceil \\
&= \left\lceil \frac{P_{tot}}{4} \right\rceil + 1
\end{aligned}
$$

Note that one additional wavelength is required to accommodate the counterclockwise set. However, if the original traffic set was unconnected and required the approach of Section 2.2.3 to turn it into a connected set (using an extra wavelength in the clockwise direction), it can share the same extra wavelength since the $\lceil PN/4 \rceil$ algorithm uses only an extra wavelength in the counterclockwise direction. In other words, for an unconnected general P-port traffic set, only a single extra wavelength is required, not two. $\square$

Here a total of at most $2\lceil (P_{tot}/4) \rceil$ converters are required. Each node i requires no more than P_i converters.

2.3.2 The $2\lceil PN/7 \rceil$ Algorithm

Again we consider the case of a connected network. The network can be either symmetric or asymmetric; again let node i have P_i ports, and define $P_{tot} = \sum_{i=1}^{N} P_i$ to be the total number of calls in the system. Find the Euler trail and list the calls in adjacent order.

By dividing the calls into adjacent sets of 7, the results of Theorem 3 can be applied to route each set using at most 2 wavelengths. Therefore a total of $\lceil 2P_{tot}/7 \rceil$. For a symmetric network, $P_{tot} = PN$, where P is the number of ports per node, and this number simplifies to $2\lceil PN/7 \rceil$. For this reason, this slightly modified algorithm is called the $2\lceil PN/7 \rceil$ algorithm.

For a connected network, a total of at most $\lfloor P_{tot}/7 \rfloor$ converters are required. Again, in Section 2.4 we will show how the wavelength assignment can be modified to distribute the converters nearly arbitrarily among all nodes. In particular, for symmetric networks, a modified wavelength assignment can be given that requires no more than $\lceil P/4 \rceil + 1$ converters per node.

2.4 The Converter-Shifting Algorithms

2.4.1 The Converter-Shifting Lemmas

In general, when a RWA algorithm gives a wavelength assignment for a traffic set, it will also specify the number of converters required at each node to support its wavelength assignment. However, this may result in inefficient use of converters since the network will have to be designed with the maximum number of converters (over all possible admissible traffic sets) at each node that the algorithm may require. For example, consider a 2-node network that sees one of two possible traffic sets, A and B. Suppose for a particular RWA traffic set A requires that node 1 have 3 converters and node 2 have 6, whereas in the RWA for traffic set B node 1 requires 6 and node 2 requires 3. Then if sets A and B are to be supported in a rearrangeably non-blocking manner, nodes 1 and 2 must both have $\max\{3, 6\} = 6$ converters, for a total of 12 converters between them, even though at most 9 converters are ever used at any given time.

In this section we provide a procedure for modifying a given wavelength assignment so that the conversion requirement can be moved arbitrarily from any node to any other node while preserving the routing of the calls. If certain criteria are met, removing one converter from a given node will require the addition of only one converter at a different node. We call this a *one-to-one* exchange. Otherwise, removing one converter from a given node will require the addition of two converters at a different node; we call this a *one-to-two* exchange.

We first define some terminology that we will find useful. A wavelength converter, when in use, converts an input wavelength to a different output wavelength. Suppose two converters are operating in the same direction (either clockwise or counterclockwise). If the output wavelength of converter 1 is the same as the input wavelength of converter 2, then we say that converter 1 is *adjacent* to converter 2, and vice versa. In particular, converter 2 is *forward adjacent* to converter 1, and converter 1 is *backward adjacent* to converter 2. Converters cannot be adjacent if they are operating in different directions.

The next two lemmas give conditions under which converters can be moved from one node to another in a one-to-one exchange. The lemmas differ in the direction a converter is shifted relative to its adjacency to the destination.

Lemma 4. *If for a given RWA a converter c_j at node j is forward adjacent to a converter c_i at node i, a modified wavelength assignment can be devised that does not require a converter at node i but may require an additional converter at node j.*

Proof. Without loss of generality, suppose the converters are operating in the clockwise direction. Call the set of all links encountered travelling from i to j in the clockwise direction the *swap set*. Let the input and output wavelengths of c_i be λ_1 and λ_2, respectively. Let the output wavelength of c_j be λ_3.

Move all traffic in the swap set on wavelength λ_1 to λ_2, and move all traffic in the swap set previously on λ_2 to λ_1. Now c_i is no longer required, since the call coming into node i on λ_1 continues on λ_1 after the swap. Also notice that calls in the swap set on λ_1 must have started at or after node i. The input wavelength of c_j becomes λ_1 after the swap, since the call which previously had been coming in on λ_2 was moved to λ_1. The output wavelength of c_j remains the same.

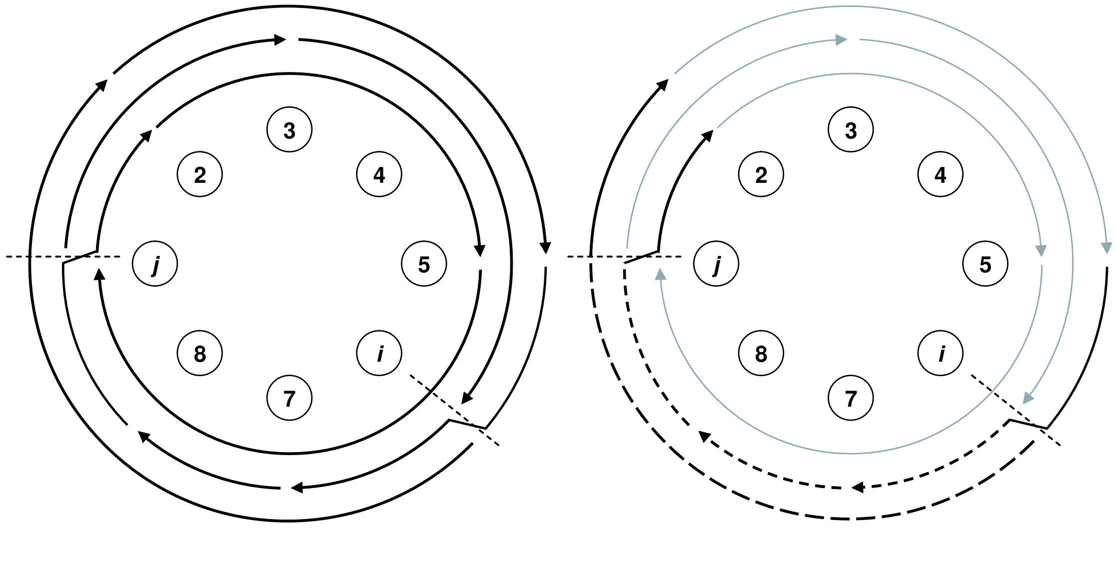
There remains one loose end to tie up. There may previously have been a call which entered node j on λ_1 and continued out on λ_1. Since after the swap this call is now entering on λ_2, an additional converter is required to convert it to λ_1 for it to continue out on λ_1 as before. Note that if the call had terminated at node j, then this converter would not be needed. □

Lemma 5. *If for a given RWA a converter c_j at node j is forward adjacent to a converter c_i at node i, a modified wavelength assignment can be devised that does not require a converter at node j but may require an additional converter at node i.*

Proof. The proof is very similar to the proof of Lemma 4. Call the set of all links encountered travelling from i to j in the clockwise direction the *swap set*. Let the input and output wavelengths of c_i be λ_1 and λ_2, respectively. Let the output wavelength of c_j be λ_3.

Move all traffic in the swap set on wavelength λ_3 to λ_2, and move all traffic in the swap set previously on λ_2 to λ_3. Now c_j is no longer required, since the call previously entering node j on λ_2 has been moved to λ_3, and may continue on λ_3 without needing a converter. The output wavelength of c_i becomes λ_3 after the swap, since the call which previously exited on λ_2 was moved to λ_3. The input wavelength of c_i remains the same.

Again there is a loose end to tie up. There may previously have been a call which entered node i on λ_3 and continued out on λ_3. Since after the swap this call



(a) (b)

Figure 2-8: (a) The original RWA of calls on the clockwise direction. Note that there is no requirement that the traffic set obey a P-port condition. Converters are used at nodes i and j. (b) The same ring, with related calls marked. Calls affected by the converter shifting are in bold, while unaffected calls are in light grey. The swap set consists of the dotted calls and parts of calls.

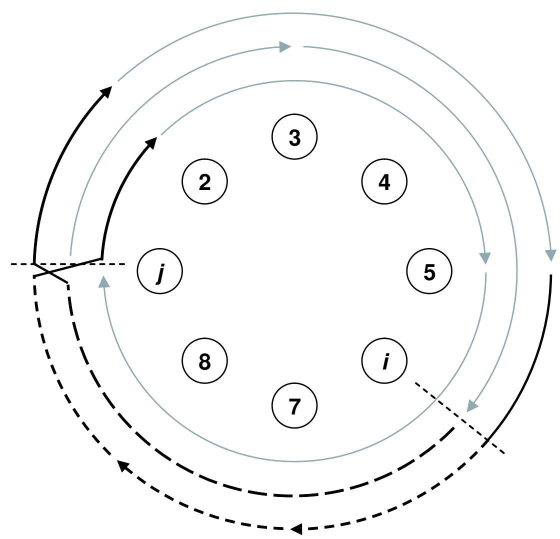


Figure 2-9: All calls or parts of calls in the short dotted lines have exchanged wavelengths with those on the long dotted lines. Note that while a converter is no longer required at node i, an extra one is now being used at node j.

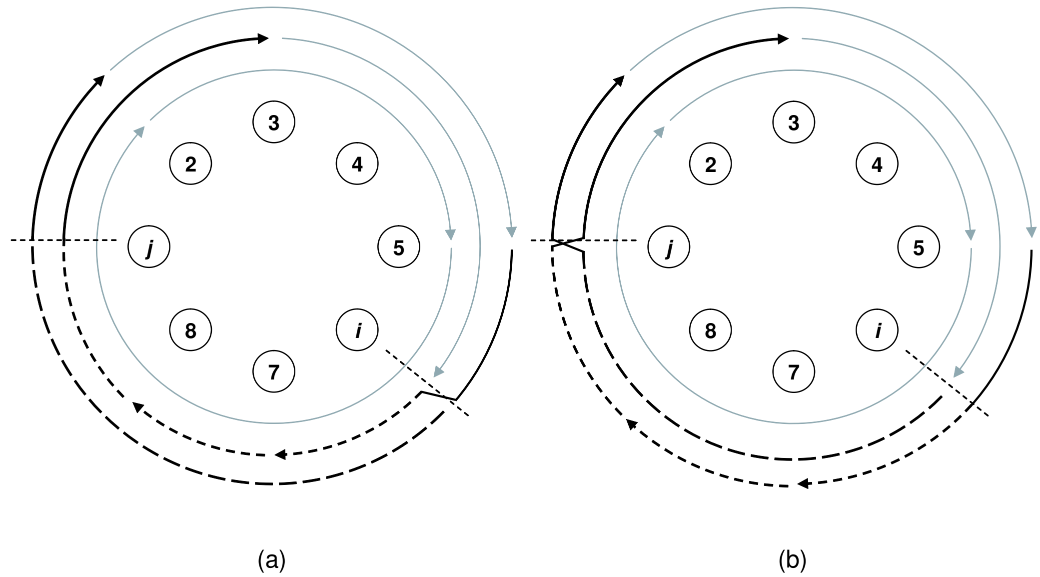


Figure 2-10: (a) The original RWA of calls on the clockwise direction. A single converters is used by node i. Calls affected by the converter shifting are in bold, while unaffected calls are in light grey. The swap set consists of the dotted calls and parts of calls. (b) All calls or parts of calls in the short dotted lines have exchanged wavelengths with those on the long dotted lines. Note that while a converter is no longer required at node i, two are used at node j.

is continuing on λ_2, an additional converter is required to convert it from λ_3 to λ_2. Note that if the call had started at node i, then again this converter would not be needed. □

An example of a one-to-one exchange of the type described in Lemma 4 is shown in Figures 2-8 and 2-9. Finally, we have a general theorem for shifting converters if no adjacent converter is available at the destination node.

Lemma 6. *If for a given RWA there does not exist any converter at node j that is adjacent to any converter at node i, a modified wavelength assignment can be devised that requires one less converter at i but may require up to two more converters at node j.*

Proof. The proof is identical to the proof of Lemma 4, except that since there is no existing adjacent converter c_j to use at node j, a new one is required. □

An example of one-to-two exchange is shown in Figure 2-10. The proofs of the preceding lemmas provide an algorithm for shifting converters from node to node.

In the following two subsections, we use the converter-shifting lemmas to first describe a method for moving all converters to a single node (typically called the *hub*), then describe a method for distributing them arbitrarily among all nodes while requiring at most one additional converter per node. The techniques used in these two examples can then be applied in a straightforward manner to implement any other configurations of interest.

2.4.2 Applications to the $\lceil PN/4 \rceil$ Algorithm

In this section, we demonstrate the use of the converter-shifting lemmas on the $\lceil PN/4 \rceil$ algorithm to create two interesting network architectures, the hub architecture and the symmetric node architecture.

1) Hub Architecture: It may be desirable to concentrate all converters at a single node, called the hub. This can be done using the converter-shifting lemmas to move all converters to the hub at a cost of at most two additional converters.

Recall that by construction at most $\lceil PN/4 \rceil - 1$ converters are used in each direction. Consider first the clockwise direction. Since by construction the converters can be traversed in adjacent order, without loss of generality we may index the converters so that converter c_i has input wavelength λ_i and output wavelength λ_{i+1}, for $i = 1, \ldots, \lceil PN/4 \rceil - 2$.

Suppose node n_h is chosen to be the hub node. According to Lemma 6, we can move c_1 to node n_h using a one-to-two exchange. Next, move converter c_2 to node n_h. Since by choice of indexing the input wavelength of c_2 is the output wavelength of c_1, by Lemma 5 it can be moved using a one-to-one exchange. Iterating through the rest of the converters, the same argument can be applied to perform one-to-one exchanges. After all exchanges are complete, there are a total of $\lceil PN/4 \rceil$ converters at the hub – one more than the previous total, due to the initial one-to-two exchange.

The same procedure can be repeated for the counterclockwise direction, resulting in an additional $\lceil PN/4 \rceil$ converters being collected at the hub. After this procedure, all conversion is now concentrated at the hub, which requires $\lceil PN/2 \rceil$ converters.

2) Symmetric Node Architecture: In other cases, we may prefer to have each node have the same number of converters. Again, this can be accomplished by using the converter-shifting lemmas to move the converters such that each node has no more than $\lceil P/2 \rceil + 1$ converters.

The procedure is as follows: first, apply the method of the previous section to create a hub architecture. There are now $\lceil PN/4 \rceil$ adjacent converters at the hub in either direction. Divide the remaining $N-1$ nodes into two sets of equal size (N odd). Call one set the *clockwise set*, and the other the *counterclockwise set*. First consider the clockwise direction. Move $\lceil P/2 \rceil$ of the converters in adjacent order to one of the $(N-1)/2$ nodes in the clockwise set. The first requires a one-to-two exchange, while all remaining converters are moved one-to-one. This places $\lceil P/2 \rceil + 1$ converters at that node. Repeat with all remaining nodes in the clockwise set. At the end of the procedure, all nodes in the clockwise set have $\lceil P/2 \rceil + 1$ converters in the clockwise direction.

Repeat this procedure with the counterclockwise set using the counterclockwise converters. This leaves all nodes in the counterclockwise set with $\lceil P/2 \rceil + 1$ converters in the counterclockwise direction. The hub itself has a total of $\lceil P/2 \rceil$ converters, half in either direction. Thus no node requires more than $\lceil P/2 \rceil + 1$ converters.

Finally, recall that the original algorithm required no more than P converters at any given node. We always retain the option of not doing any converter shifting if $\lceil P/2 \rceil + 1 > P$. (As a side note, we point out that the only time this occurs is at $P = 1$.) Therefore the final result is that the number of converters required per node is given as $\min\{\lceil P/2 \rceil + 1, P\}$.

2.4.3 Applications to the $2\lceil PN/7 \rceil$ Algorithm

In this section, we demonstrate the use of the converter-shifting lemmas on the $2\lceil PN/7 \rceil$ algorithm to again create a hub and symmetric node architectures.

1) Hub Architecture: The converter-shifting lemmas can be used to move all converters to a single node. For the $2\lceil PN/7 \rceil$ algorithm, converter adjacency is not guaranteed, and hence redistribution requires one-to-two exchanges. Hence the hub has at most $2\lfloor PN/7 \rfloor$ converters.

2) Symmetric Node Architecture: The converter-shifting lemmas can also be used to move converter requirements to ensure that each node requires no more than $\lceil P/4 \rceil$ converters.

The procedure is as follows. Locate the nodes which require more than $\lceil P/4 \rceil$ converters. Define these nodes to be members of the set R requiring relocation. Consider the first converter in the set R. Locate a node not contained in R which currently has fewer than $\lceil P/4 \rceil$ converters, and move it to that node. We call this the *relocation step*, which is at worst a one-to-two exchange. Repeat the relocation step until the number of converters at that node drops to $\lceil P/4 \rceil$. Remove that node from the set R, then move onto the next node in R and repeat, until the set R is empty.

We claim that we can always perform the relocation step for all nodes in R; that is, we never run out of nodes with fewer than $\lceil P/4 \rceil$ converters while there remain nodes in R with converters which need to be relocated. This claim is formalized in the following theorem.

Theorem 6. *Define the* excess demand D *for converters to be the sum of the minimum number of converters which need to be removed from each node so that the number of the converters at the node does not exceed* $\lceil P/4 \rceil$*. Define the* excess capacity C *to be the sum of the maximum number of converters which could be added at each node without exceeding* $\lceil P/4 \rceil$*. Denote by* X_i *the quantity of converters required at node* i *by a given RWA. Mathematically, these quantities are related by:*

$$
\begin{aligned}
D &= \sum_{i=1}^{N} max\left(X_i - \left\lceil \frac{P}{4} \right\rceil, 0\right) \\
&= \sum_{i \in R} \left(X_i - \left\lceil \frac{P}{4} \right\rceil\right) \\
C &= \sum_{i=1}^{N} max\left(\left\lceil \frac{P}{4} \right\rceil - X_i, 0\right) \\
&= \sum_{i \in R^C} \left(\left\lceil \frac{P}{4} \right\rceil - X_i\right)
\end{aligned}
$$

where R^C denotes the complement of R; i.e., R^C is composed of those nodes not contained in R.

Then the theorem asserts that

$$2D \leq C$$

Proof. Index the nodes $n_1, \ldots, n_N$ such that $n_1, \ldots, n_j$ all have more than $\lceil P/4 \rceil$ converters, while the remaining nodes $n_{j+1}, \ldots, n_N$ do not. By this choice of indexing, the set R is composed of the nodes $\{n_1, \ldots, n_j\}$. The expressions for D and C can be written as:

$$
\begin{aligned}
D &= \sum_{i=1}^{j} \left(X_i - \left\lceil \frac{P}{4} \right\rceil\right) \\
&= \left(\sum_{i=1}^{j} X_i\right) - j \left\lceil \frac{P}{4} \right\rceil \\
C &= \sum_{i=j+1}^{N} \left(\left\lceil \frac{P}{4} \right\rceil - X_i\right) \\
&= (N - j) \left\lceil \frac{P}{4} \right\rceil - \left(\sum_{i=j+1}^{N} X_i\right)
\end{aligned}
$$

To prove the theorem, we must show that $C - 2D \geq 0$. To see this, begin with

$$
\begin{aligned}
C - 2D &= (C - D) - D \\
&= (N - j)\left\lceil \frac{P}{4} \right\rceil - \sum_{i=j+1}^{N} X_i \\
&\quad - \sum_{i=1}^{j} X_i + j\left\lceil \frac{P}{4} \right\rceil - D \\
&= \left(N\left\lceil \frac{P}{4} \right\rceil - \sum_{i=1}^{N} X_i \right) - D \\
&\geq \left(N\left\lceil \frac{P}{4} \right\rceil - \left\lfloor \frac{PN}{7} \right\rfloor \right) - D \\
&\geq \left(\frac{PN}{4} - \frac{PN}{7} \right) - D \\
&= \frac{3}{28} PN - D
\end{aligned}
$$

where the first inequality arises from the fact that the total number of converters required $\sum_{i=1}^{N} X_i \leq \lfloor PN/7 \rfloor$, and the second is from the removal of the floor and ceiling functions.

We next need to determine an upper bound on the excess demand D. To develop this bound, we formulate an equivalent problem involving balls and jars. Consider the problem of distributing $\lfloor PN/7 \rfloor$ balls into N jars, where each jar can hold at most P balls stacked vertically, in order to maximize the total number of balls in the jars exceeding a height of $\lceil P/4 \rceil$. This is illustrated in Figure 2-11. The balls correspond to converters, the jars to nodes, and the number of balls which exceed height $\lceil P/4 \rceil$ is equal to the quantity D.

An algorithm for maximizing the number of balls placed which exceed height $\lceil P/4 \rceil$ is to begin at the first jar, fill it with as many balls as possible, move to the next jar, and repeat. Then the number of jars required is $\frac{\lfloor PN/7 \rfloor}{P} \leq \frac{PN/7}{P} = N/7$, and each jar has an excess capacity of at most $P - \lceil P/4 \rceil \leq P - P/4 = 3P/4$. Therefore the excess demand is at most $D \leq (P/7)(3P/4) = 3PN/28$.

Using this inequality in Equation (2.1), we then have

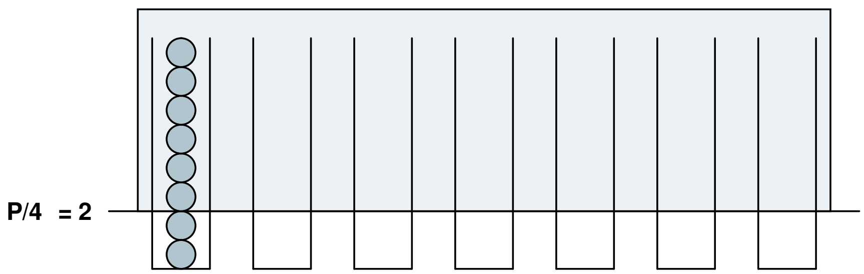


Figure 2-11: An example of the ball distribution problem. The excess capacity (represented by balls falling in the shaded area) is maximized by filling each jar as much as possible before moving onto the next jar.

$$
\begin{aligned}
C - 2D &\geq \frac{3}{28}PN - \frac{3}{28}PN \\
&= 0
\end{aligned}
$$

which proves the theorem. □

A direct corollary of this theorem is that converters can be equally distributed so that no node needs more than $\lceil P/4 \rceil + 1$ converters. The one-to-two shifting is the reason for the extra "+1" term. In the worst case, it is possible that a converter may be shifted to a node outside the set R which prior to the shifting had $\lceil P/4 \rceil - 1$ converters; in this case, adding two additional converters gives it a total of $\lceil P/4 \rceil + 1$.

Again, since the original algorithm required no more than P converters per node, we retain the option of doing no shifting if $\lceil P/4 \rceil + 1 > P$. Therefore in the final assessment the number of converters required per node is $\min\{\lceil P/4 \rceil + 1, P\}$.

2.5 Chapter Summary

In this chapter, we considered the problem of implementing all virtual topologies on an N-node P-port network in a rearrangeably non-blocking fashion while trying to minimize the number of wavelengths and converters required. We show that for symmetric P-port networks, a lower bound on the number of wavelengths is $\lceil PN/4 \rceil$. We present an algorithm which achieves this lower bound by using $\lceil PN/4 \rceil$ wavelengths for connected topologies while using a total of no more than $\lceil PN/2 \rceil - 2$ converters. We also present a second algorithm which uses $2\lceil PN/7 \rceil$ wavelengths but requires fewer converters, a total of no more than $\lfloor PN/7 \rfloor$. The first algorithm achieves the minimum number of wavelengths required, while the second uses more wavelengths but greatly reduces the number of converters used. We also show how to turn the problem of implementing an unconnected traffic set into a modified problem of implementing a connected set by using a single additional wavelength. We then extend the results to general P-port networks, where we allow the number of ports P_i at each node i to vary, and show that for such networks the $\lceil PN/4 \rceil$ algorithm requires no more than $\lceil \sum_i P_i/4 \rceil + 1$ wavelengths for connected and unconnected traffic sets. A similar extension for the $\lceil PN/7 \rceil$ algorithm shows that it requires only $\lceil \sum_i P_i/7 \rceil + 1$ wavelengths.

Finally, we demonstrate a method for changing wavelength assignments to move converters arbitrarily from one node to another. If certain conditions are met, we show that this exchange is one-to-one; otherwise, the exchange is one-to-two. We also show how to apply this method to both the $\lceil PN/4 \rceil$ and $2\lceil PN/7 \rceil$ algorithms. For symmetric P-port networks, we demonstrate a hub topology for the $\lceil PN/4 \rceil$ algorithm which uses $\lceil PN/2 \rceil$ converters at the hub and no converters elsewhere, and a symmetric node topology which uses $\lceil P/2 \rceil + 1$ converters at each node. We also give a hub topology for the $2\lceil PN/7 \rceil$ algorithm which uses $2\lceil PN/7 \rceil$ converters at the hub and no converters elsewhere, and a symmetric node topology which uses at most $\lceil P/4 \rceil + 1$ converters at each node. For asymmetric networks, the expressions are the same except that $P_{tot} = \sum_i P_i$ replaces PN.

It is worth comparing the worst-case wavelength requirement to the wavelength requirement for static and uniform all-to-all traffic. In all-to-all uniform traffic, each node communicates with every other node. For N odd, this requires $(N^2 - 1)/8$ wavelengths [35, 12]. In our terminology, all-to-all traffic belongs to the admissible set of an N-node network with $N-1$ ports, which have a worst-case bound of $N(N-1)/4$

wavelengths. Thus designing a network to support $P = N-1$ calls per node uses twice as many wavelengths as a uniform all-to-all design. However, the P-port traffic model provides significantly more flexibility than the uniform all-to-all model. Furthermore, an argument given in [24] can be used to show that a large number of topologies require the lower bound of $\lceil PN/4 \rceil$ wavelengths for the P-port case, showing that this bound is not inflated to support only a small number of worst-case scenarios.

2.6 Chapter Appendix

In this section we consider the number of wavelengths required by the $\lceil N/4 \rceil$ algorithm in the counterclockwise direction for the case of $k = \lfloor \frac{N^2}{4L} \rfloor$. Recall that the number of hops of traffic in the counterclockwise set was given by

$$\begin{aligned} D_W &= (N-k) \cdot (N - \hat{L}) \\ &\leq (N-k) \cdot (N - \bar{L}) \\ &= \left(N - \left\lfloor \frac{N^2}{4\bar{L}} \right\rfloor \right) \cdot (N - \bar{L}) \end{aligned}$$

Consider the maximization of the right-hand side; that is, the function

$$f(L) = \left(N - \left\lfloor \frac{N^2}{4L} \right\rfloor \right) \cdot (N - L) \tag{2.1}$$

The number of nodes N must obviously be an integer, and we can also deduce that the average hop length $\bar{L}$ is also integer. To see this, recall that we assumed the traffic set was connected. This implies that, starting at any node, we can proceed in adjacent order through all the calls in the clockwise direction and return to the same node. Thus, the total number of hops of traffic in the clockwise direction must be an integer multiple of N. Therefore the average hop length, which we obtain by dividing the total hop length by the number of nodes N, must also be integer.

For the proof we will also only consider the case where N is even. There is no loss of generality because in all cases of practical interest, this assumption holds. To see this, consider a ring network with N odd. We can add a fictitious node $N+1$ to make the total number of nodes even. We alter the traffic set by arbitrarily picking any call from the original traffic set. Suppose this call is from node n_i to n_j, denoted by (n_i, n_j). We remove this call from the traffic set and replace it by two calls (n_i, x) and (x, n_j). Observe that this new traffic set, over the $(N+1)$-node ring, is now a maximal single-port traffic set. It also retains connectedness.

The number of wavelengths required to route the new traffic set using the $\lceil N/4 \rceil$ algorithm is $\lceil (N+1)/4 \rceil$. Since for N odd $\lceil (N+1)/4 \rceil = \lceil N/4 \rceil$, no additional wavelengths are required by this procedure. Once routes have been found for all calls, remove the fictitious node x. Then use the route determined for the calls (n_i, x) and (x, n_j) to route the original call (n_i, n_j). This shows that it is sufficient to consider the case of only N even, because it allows us to also perform RWA for N odd without using any additional wavelengths.

Returning our attention to the function $f(L)$, we are interested in finding an upper bound. The goal will be to show that the total hops of traffic is no greater than $N^2/4$, and by combining this with the fact that each wavelength provides N hops of traffic capacity, we will also prove that the counterclockwise set requires no more than $\lceil N/4 \rceil$ wavelengths.

The proof will proceed by showing the following two relations:

1. For all $k \in \{1, \ldots, N/2\}$, $f(\frac{N}{2}) \geq f(\frac{N}{2} - k)$

2. For all $k \in \{1, \ldots, N/2\}$, $f(\frac{N}{2}) \geq f(\frac{N}{2} + k)$

Together, the two relations show that $f(L)$ is maximized at $L = N/2$. Since $f(N/2) = N^2/4$, this leads to the desired result.

We proceed with showing the first inequality. We first introduce a useful lemma, followed by the proof of the theorem.

Lemma 7. *For $k \in \{1, \ldots, N/2\}$ and $N/2$ integer,*

$$\left\lfloor \frac{N^2}{4(\frac{N}{2}-k)} \right\rfloor \geq \frac{N}{2} + k$$

Proof. We begin by showing

$$\begin{aligned}
\frac{N^2}{4(\frac{N}{2}-k)} &= \frac{N^2}{2N(1-\frac{2k}{N})} \\
&= \frac{N}{2(1-\frac{2k}{N})}
\end{aligned}$$

Using this result, we can then also show that

$$\begin{aligned}
\frac{N^2}{4(\frac{N}{2}-k)} - \frac{N}{2} &= \frac{N}{2(1-\frac{2k}{N})} - \frac{N}{2} \\
&= \frac{k}{1-\frac{2k}{N}} \\
&> k
\end{aligned}$$

and therefore

$$\frac{N^2}{4(\frac{N}{2}-k)} > \frac{N}{2} + k$$

Taking the floor of both sides,

$$\begin{aligned}
\left\lfloor \frac{N^2}{4(\frac{N}{2}-k)} \right\rfloor &\geq \left\lfloor \frac{N}{2} + k \right\rfloor \\
&= \frac{N}{2} + k
\end{aligned}$$

where the last step follows from the fact that both $N/2$ and k are integers. This proves the lemma. □

Theorem 7. *For* $k \in \{1, \ldots, N/2\}$ *and* $N/2$ *integer,*

$$f\left(\frac{N}{2}-k\right) \leq f\left(\frac{N}{2}\right)$$

Proof. Beginning at the definition of $f(L)$, we have:

$$\begin{aligned} f\left(\frac{N}{2}-k\right) &= \left(N - \left\lfloor \frac{N^2}{4(\frac{N}{2}-k)} \right\rfloor\right) \\ &\quad \cdot \left(N - \left(\frac{N}{2}-k\right)\right) \\ &\leq \left(N - \left(\frac{N}{2}+k\right)\right) \cdot \left(\frac{N}{2}+k\right) \end{aligned}$$

where the last inequality was obtained using Lemma 7. Continuing, a few additional algebraic steps gives us

$$\begin{aligned} f\left(\frac{N}{2}-k\right) &\leq \left(\frac{N}{2}-k\right) \cdot \left(\frac{N}{2}+k\right) \\ &= \frac{N^2}{4} - k^2 \\ &\leq \frac{N^2}{4} \end{aligned}$$

Since $f\left(\frac{N}{2}\right) = \frac{N^2}{4}$, this shows that

$$f\left(\frac{N}{2}-k\right) \leq f\left(\frac{N}{2}\right)$$

which proves the theorem. □

The proof of the second inequality parallels the development of the proof of the first very closely. Again, a helpful lemma will first be developed before the theorem is presented.

Lemma 8. *For $k \in \{1, \ldots, N/2\}$ and $N/2$ integer,*

$$\left\lfloor \frac{N^2}{4(\frac{N}{2}+k)} \right\rfloor \geq \frac{N}{2} - k$$

Proof. We begin by observing

$$\begin{aligned} \frac{N^2}{4(\frac{N}{2}+k)} &= \frac{N^2}{2N(1+\frac{2k}{N})} \\ &= \frac{N}{2(1+\frac{2k}{N})} \end{aligned}$$

Using the above, we have

$$\begin{aligned} \frac{N}{2} - \frac{N^2}{4(\frac{N}{2}+k)} &= \frac{k}{1+\frac{2k}{N}} \\ &< k \\ \Rightarrow \frac{N^2}{4(\frac{N}{2}+k)} &> \frac{N}{2} - k \end{aligned}$$

Taking the floor of both sides,

$$\begin{aligned} \left\lfloor \frac{N^2}{4(\frac{N}{2}+k)} \right\rfloor &\geq \left\lfloor \frac{N}{2} - k \right\rfloor \\ &= \frac{N}{2} - k \end{aligned}$$

where the last line follows from the fact that both $N/2$ and k are integers. This proves the lemma.

□

Theorem 8. *For $k \in \{1, \ldots, N/2\}$ and $N/2$ integer,*

$$f\left(\frac{N}{2}+k\right) \leq f\left(\frac{N}{2}\right)$$

Proof. Beginning at the definition of $f(L)$ and applying Lemma 8, we have:

$$\begin{aligned} f\left(\frac{N}{2}+k\right) &\leq \left(N-\left(\frac{N}{2}-k\right)\right)\cdot\left(\frac{N}{2}-k\right) \\ &\leq \frac{N^2}{4}-k^2 \\ &\leq \frac{N^2}{4} \end{aligned}$$

Since $f\left(\frac{N}{2}\right)=\frac{N^2}{4}$, this shows that

$$f\left(\frac{N}{2}+k\right) \leq f\left(\frac{N}{2}\right)$$

which proves the theorem. □

Chapter 3

Channel Access and Optical Bypass

In Chapter 2 we considered routing and wavelength assignment, and used total number of wavelengths as the cost metric. As previously mentioned, this is sensible if all wavelengths are accessed at all nodes – reducing the number of wavelengths therefore reduces the hardware costs as well.

However, in a medium to large network, the amount of traffic (in wavelengths) that each node contributes is typically a small fraction of the total number of wavelengths in the fiber. It therefore becomes sensible to consider allowing the majority of the wavelengths entering a node to simply optically bypass the node, and tap only a sufficiently large fraction of the wavelengths to add outgoing calls and remove incoming ones. So-called bypass wavelengths then become very inexpensive to manage from a hardware perspective, since they will only be processed at a very limited subset of access nodes. Under this architecture, we impose a limited-accessibility constraint to the wavelength set (making the RWA problem more complex) in order to reduce overall hardware complexity.

In this chapter, we will investigate this optical bypass problem. Specifically, we focus on locally-accessible wavelengths as the primary cost criterion, and determining the minimum number of locally-accessible wavelengths required. Total number of wavelengths should also be a (secondary) consideration, and we will investigate architectures that allow the majority of wavelengths to optically bypass most nodes. The details of the traffic model are first discussed in Section 3.1, followed by a description of the optimization criteria.

3.1 System Model

We consider three topologies: the ring, the two-dimensional torus, and the tree. This will allow for an understanding of bypass architectures in a variety of environments.

In a ring topology, adjacent nodes are connected by two fibers: one supporting wavelengths travelling in the clockwise direction, the other supporting wavelengths in the counterclockwise direction. The two fibers are represented by a single bidirectional link, where each link can support calls travelling in both directions on every wavelength. If we number the nodes from 0 to $N-1$, then each node i is connected to nodes $i \oplus 1$ and $i \ominus 1$, where $\oplus$ and $\ominus$ denote addition and subtraction modulo N, respectively.

A two-dimensional (2D) torus is essentially a rectangular topology where each node is a member of two rings. The total number of nodes N must be the product of two integers R and C, and the resultant topology is known as an $R \times C$ torus. Examples of torus topologies are given in Section 3.3.

A tree topology consists of nodes joined by bidirectional fiber links such that the undirected graph underlying the network must be connected and contain no cycles [3].

We will consider networks with and without wavelength converters. A wavelength converter, if available at a given node, can be used to switch a call arriving to that node on one wavelength onto a different wavelength departing the node. If no conversion is employed, a call passing through a node on one wavelength must exit the node on the same wavelength.

A traffic matrix or traffic set consists of a set of calls that need to be set up in the network. Each call consists of a source and destination pair and is assumed to occupy a full wavelength. A traffic set is *connected* if the directed graph corresponding to the set of source-destination pairs is connected. In the P-port model, the number and types of calls which the traffic set may contain are based on the port limitations at each node. Each node i is considered to have an equal number of ports P_i for transmitting and receiving, and hence may be the source and destination of at most P_i calls within any given traffic set. This traffic model is natural because it places constraints on the calls which are based on the actual equipment constraints at each node. A traffic set which obeys these constraints is said to be *admissible*.

The dynamic nature of the traffic is modelled by transitions from one admissible traffic set to another over time. The RWA algorithm must provision enough resources to support *any* admissible traffic set, so that regardless of the evolution of the network state over time, no calls are blocked. For this reason, the model is called *non-blocking*. We further allow existing calls to be rearranged between transitions; hence the term *rearrangeably non-blocking*.

In general, as in Chapter 2, each node i can have a different number of ports P_i. The special case where all nodes have the same number of ports $P_i = P$ is called *symmetric P-port traffic*. A special case of symmetric P-port traffic is the *single-port traffic* case, where $P = 1$ for every node.

3.1.1 Objective Function

For the purposes of the RWA problem, we can group the wavelengths into two bands: a *local* band, consisting of wavelengths that can be accessed by all nodes, and a *bypass* band, consisting of wavelengths that can be accessed only by a few designated *hub nodes*. The bypass band can therefore bypass the majority of the nodes in the network. The bypass band can be viewed as a set of "highways" that can only be accessed via the hub nodes, which serve as access ramps by using wavelength conversion to translate incoming calls on a local wavelength to a bypass wavelength and vice versa, as necessary.

There are several advantages to a banding approach. One is cost savings. Figure 3-1 gives an example of an optical add-drop multiplexer (OADM) in a network with no banding. In a system with W total wavelengths, all W wavelengths would have

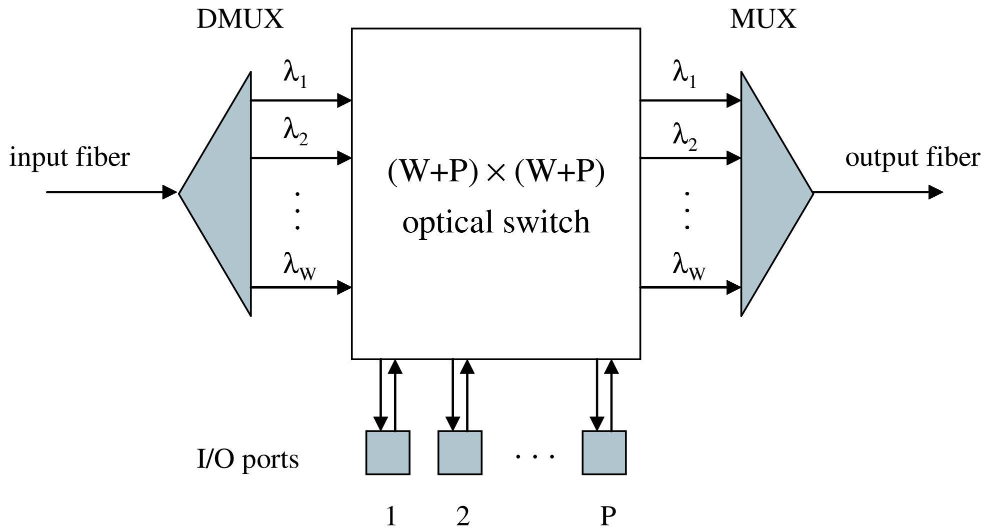


Figure 3-1: An OADM architecture. Note that a (W+P) × (W+P) switch is required, and that the size of this switch increases with the number of wavelengths.

to be switched. Figure 3-2 shows a system where the wavelengths are divided into a local band of k wavelengths, and a bypass band of $W - k$ wavelengths. In this case, only the smaller local band of k wavelengths is switched. Another benefit is that the wavelength demultiplexers can be simpler: the first, coarse DMUX need only separate out two large bands, while the second, finer DMUX has a smaller band to work with (only the local wavelengths). Finally, by allowing wavelengths in the bypass band to avoid processing at non-hub nodes altogether, the bypass band can either avoid the switch (thus not suffering power losses due to switching which would reduce the reach of the lightpaths), or be placed in a separate fiber entirely. Such a separate fiber would need to be connected only to the hub nodes and could physically bypass all other nodes entirely.

The potential disadvantage, of course, is that by restricting the access of some of the wavelengths to only the hub nodes, we are imposing additional constraints onto the RWA problem that may decrease the efficiency of the algorithms and necessitate the provisioning of more overall wavelengths.

In this chapter, we provide novel RWA algorithms with and without bypass for a variety of topologies. We demonstrate for each case optimal or efficient wavelength

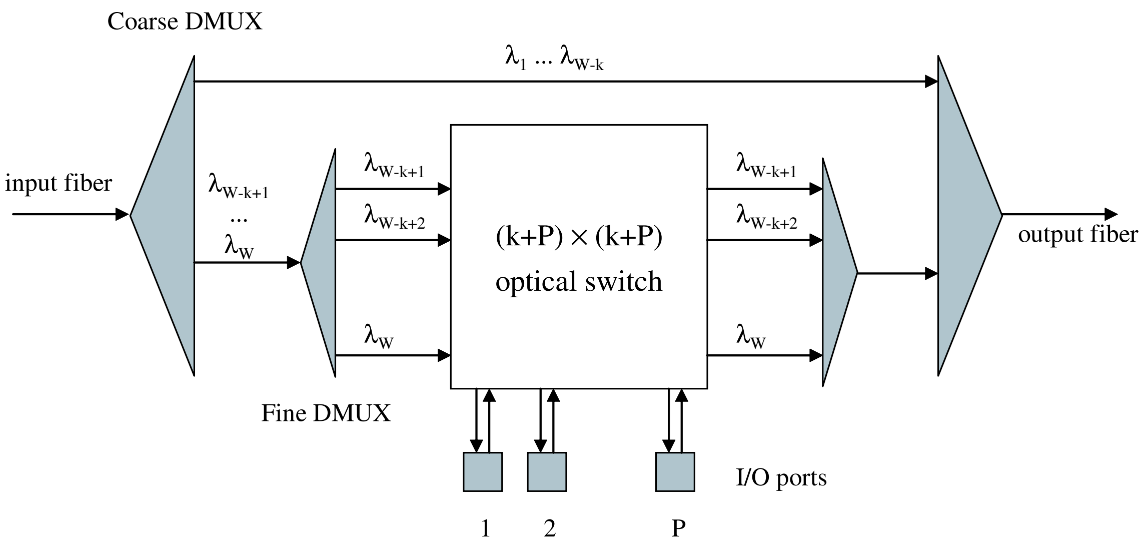


Figure 3-2: An OADM architecture with bypass. Note that the switch size depends only on the number of wavelengths k entering the node, not the total number W.

usage through comparisons with theoretical lower bounds, and demonstrate ways for using banding to further simplify node complexity. In Section 3.2 we cover RWA and banding algorithms for ring networks. A similar approach using ring embeddings to extend these results to torus and tree topologies can also be derived but is omitted here for brevity. Table 5.1 provides a summary of the wavelength usage of the algorithms to be presented. Typically, in each section, a RWA algorithm which achieves the lower bound on the total number of wavelengths used is presented, followed by a banding algorithm which reduces the number of local wavelengths at the expense of the overall total number of wavelengths or wavelength converters.

3.2 Ring Networks

3.2.1 Rings Without Conversion

In this section, RWA algorithms which minimize the total number of wavelengths required to support P-port traffic in an N-node ring are considered. We first establish a bound on the minimum number of wavelengths required without conversion. Recall

Table 3.1: Algorithm performance summary

Type	Algorithm	λ's (local)	λ's (total)
ring	lower bound	$\lceil PN/8 \rceil$	$\lceil PN/4 \rceil$
	$\lceil PN/4 \rceil$ alg.	$\lceil PN/4 \rceil$	$\lceil PN/4 \rceil$
	ring banding	$\lceil PN/8 \rceil$	$\lceil 3PN/8 \rceil$
torus	low. bound ($R \geq C$)	–	$\lceil PR/4 \rceil$
	TERA ($R \geq 2C$)	$\lceil PR/4 \rceil$	$\lceil PR/4 \rceil$
	banding ($R \geq 2C$)	$\lceil PR/8 \rceil$	$\lceil 3PR/4 \rceil$
tree	lower bound (binary)	–	$\lceil PN/2 \rceil$
	banding	$\lceil PN/4 \rceil$	$\lceil PN/2 \rceil$

that under the P-port traffic model, any traffic set in which each node i sends and receives no more than P_i calls is admissible; as such, no call belonging to such a set may be blocked. Therefore sufficiently many wavelengths must be provisioned to support all calls within any given admissible traffic set. In [24] it was shown that for a ring with $N > 7$ nodes, there exists an admissible traffic set which requires at least $\lceil PN/3 \rceil$ wavelengths to support it if no wavelength conversion is available. A RWA algorithm was also described that always uses no more than this minimum number of wavelengths for any admissible traffic set. The algorithm was based on the following observation:

Lemma 9. *Define a set of calls to be* adjacent *if the destination of the first call is the source of the second call, the destination of the second call is the source of the third call, and so forth. Given any two adjacent calls, if the calls cannot fit on a single bidirectional wavelength in the clockwise direction, they must fit onto a single wavelength in the counterclockwise direction.*

A graphical proof is shown in Figure 3-3. The algorithm in [24] divides the calls into sets of three adjacent calls each, and observes that as a consequence of Lemma 9 each set could fit on a single wavelength. To do this, determine the direction in which the first two adjacent calls can share a wavelength, and route them in that direction using a wavelength. Route the third call in the opposite direction using the same wavelength.

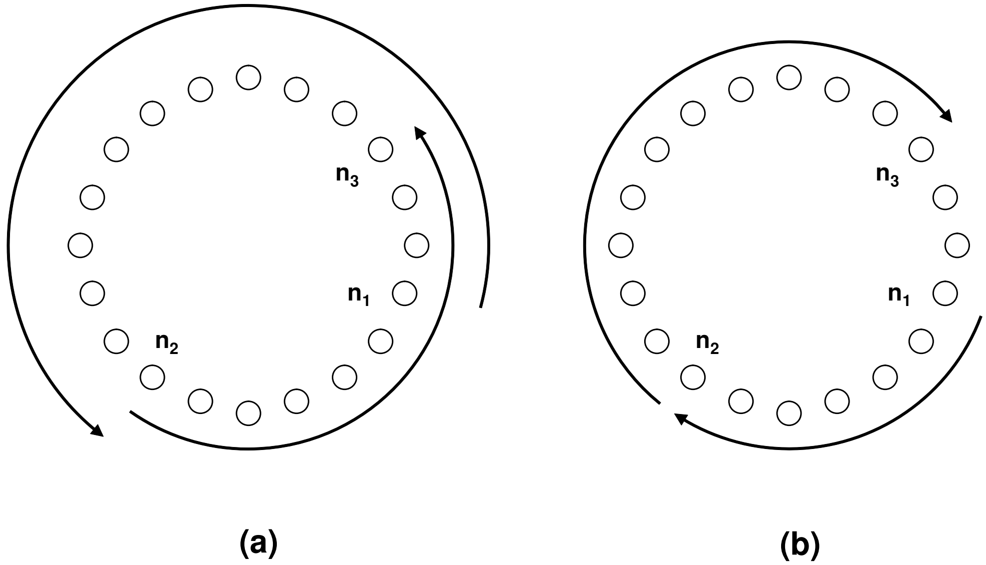


Figure 3-3: (a) Two calls, (n_1, n_2) and (n_2, n_3), which cannot fit on a single counterclockwise wavelength since the second call overlaps from n_1 to n_3 with the first call. (b) The same two calls can use a single wavelength in the clockwise direction, because the second call must reach destination n_3 in the clockwise direction before encountering the source n_1 of the first call.

3.2.2 Rings With Conversion

We next consider P-port bidirectional rings with conversion, again under the objective of minimizing the number of wavelengths. The cut-set bound can be used to provide a lower bound for symmetric P-port traffic. Consider a cut of two links which divides the ring in equal halves of $N/2$ nodes each. A worst-case admissible P-port traffic set can be constructed where each node on the left half of the cut sends all P units of traffic to some node on the right half. This means that $PN/2$ units of traffic would traverse 2 links, requiring a minimum of $\lceil PN/4 \rceil$ wavelengths.

An algorithm was provided in [6] called the $\lceil PN/4 \rceil$ algorithm which provides a RWA for any P-port traffic set using no more than $\lceil PN/4 \rceil$ wavelengths. This is optimal in the sense that it meets the lower bound on wavelength usage. It requires a total of $\lceil PN/2 \rceil$ wavelength converters, which can be arbitrarily located within the ring. A detailed description of the algorithm is given in [6]; for the purposes of our discussion here it suffices to know that such an algorithm exists. This algorithm will be used later in the paper to assist in performing RWA on other topologies.

A corollary is that for the special case of $N = 4$ and $P = 1$, only a single wavelength is required, and no wavelength converters are used. This can be extended to the case of arbitrary P by noting that any P-port set can be decomposed into P single-port sets, and each set routed individually. Hence for $N = 4$ the lower bound of P wavelengths can be achieved without conversion.

3.2.3 Ring Banding Bound

In this section, we consider maximizing the amount of bypass using banding. Recall that wavelengths are divided into two sets of adjacent wavelengths known as the local band and the bypass band. Wavelengths in the local band, known as local wavelengths, can be accessed by any node; wavelengths in the bypass band, known as bypass wavelengths, can be accessed only by a few special nodes known as hub nodes. Any non-hub node is termed a local node. As described in Section 3.1.1, maximizing the size of the bypass band corresponds to reducing the overall cost of the network.

In this section, we assume symmetric P-port traffic and derive a lower bound on the size of the local band for a fixed number of hubs h within an N-node ring. In general, the number of local wavelengths required decreases as the number of hubs increases. It is possible to lower-bound the number of local wavelengths required for a fixed number of hubs using a cut-set bound.

The hub nodes divide the ring into sections, each consisting of a number of local nodes located between two consecutive hub nodes. Suppose the smallest such section consists of N_S local nodes. A traffic set exists where each node within that section sends all P calls to nodes outside that section. Imagine a cut consisting of the two links at the edge of that section. There are $P \cdot N_S$ units of traffic travelling across two links, and so a minimum of $\lceil P \cdot N_S/2 \rceil$ wavelengths are required. Furthermore, these must all be local wavelengths, since none of the local nodes can access bypass wavelengths. To obtain the tightest such bound, we maximize N_S by distributing the hubs symmetrically within the ring, resulting in $N_S = \lceil N/h \rceil - 1$ and a lower bound of $\lceil P(\lceil N/h \rceil - 1)/2 \rceil$. This situation is illustrated in Figure 3-4 for the case of $h = 4$.

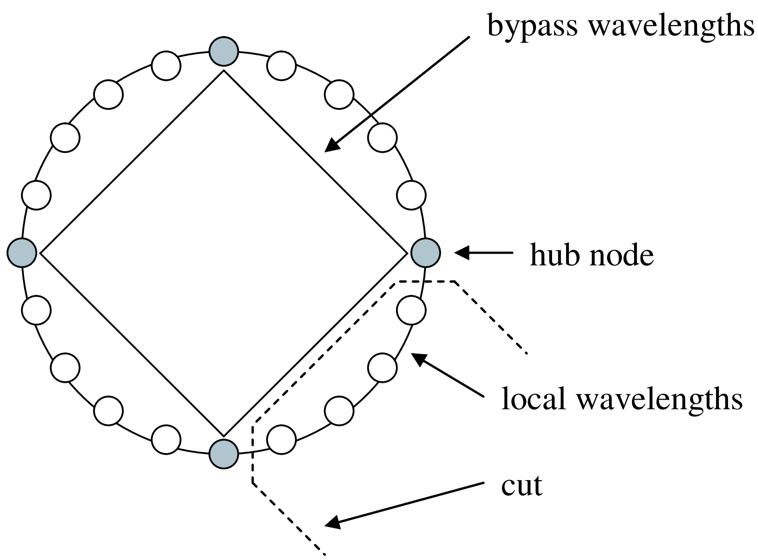


Figure 3-4: A dual ring topology, equivalent to a 20-node, 4-hub ring with some local and some bypass wavelengths. The shaded nodes are hubs.

3.2.4 Banding and Bypass on Rings

In this section, an algorithm called the *ring banding algorithm* that minimizes the number of wavelengths in the local band is described. In general, there exists an entire class of ring banding algorithms depending on the number of hubs in the ring; however, from a single instance of the algorithm it is straightforward to extrapolate to any other variation since the algorithms are all very similar. Therefore in this section we focus primarily on the special case of a 4-hub ring banding algorithm. A 4-hub architecture has the added advantage that routing traffic between the four hubs does not require wavelength conversion to achieve optimal wavelength efficiency, as noted in the preceding section.

For $h = 4$ hubs, the lower bound is $\lceil P(\lceil N/h \rceil - 1)/2 \rceil = \lceil P(\lceil N/4 \rceil - 1)/2 \rceil$ wavelengths. To obtain some intuition about how large this local band is, consider the case where N is a multiple of 4; under this assumption, this minimum number becomes $\lceil \frac{P(N-4)}{8} \rceil$. This establishes that at least half of all the wavelengths must be local wavelengths. (Recall that $\lceil PN/4 \rceil$ total wavelengths are required.)

To reduce the number of local wavelengths required, consider a topology where the

4 hubs are distributed symmetrically within the ring. The ring banding algorithm gives each call a route and wavelength assignment using a three-step process, as follows:

1. Starting from the source node, the call travels to the nearest hub. This route uses a local wavelength and is static since the nearest hub node for any given source node is fixed.

2. From that hub, it travels via a bypass wavelength to the hub closest to the destination node. This routing is dynamic since the source and destination of the call are variable.

3. Finally, the call proceeds from that hub to the destination node via a local wavelength, again by a static route.

We first prove that Steps 1 and 3 use no more than the minimum number of local wavelengths. Then we provide an RWA to dynamically route all the calls in Step 2 using as few bypass wavelengths as possible.

Consider the local wavelength usage. The hubs partition local nodes in the ring into four quarters, each of which contains no more than $\lceil N/4 \rceil - 1$ local nodes. Assign each node within the quarter P bidirectional wavelengths for communication with the hub closest to it. The forward direction, from the node to the hub, is used in Step 1; the reverse direction, from the hub to the node, is used in Step 3. Since half the nodes in each quarter communicate with the hub node on one side and half with the other, each bidirectional wavelength can be shared by two local nodes. This is illustrated for a 16-node ring in Figure 3-5. In total, $\lceil P(\lceil N/4 \rceil - 1)/2 \rceil$ local wavelengths are required by this scheme. Note that this meets the lower bound on local wavelengths, and corresponds to roughly half of the total number of wavelengths used by the $\lceil PN/4 \rceil$ algorithm.

Next we consider the dynamic routing of calls between hubs in Step 2. Each hub is responsible for sending and receiving all calls belonging to the $\lceil N/4 \rceil$ nodes closest to it (including itself) to and from other hubs. Therefore, for Step 2 each of the 4 hubs

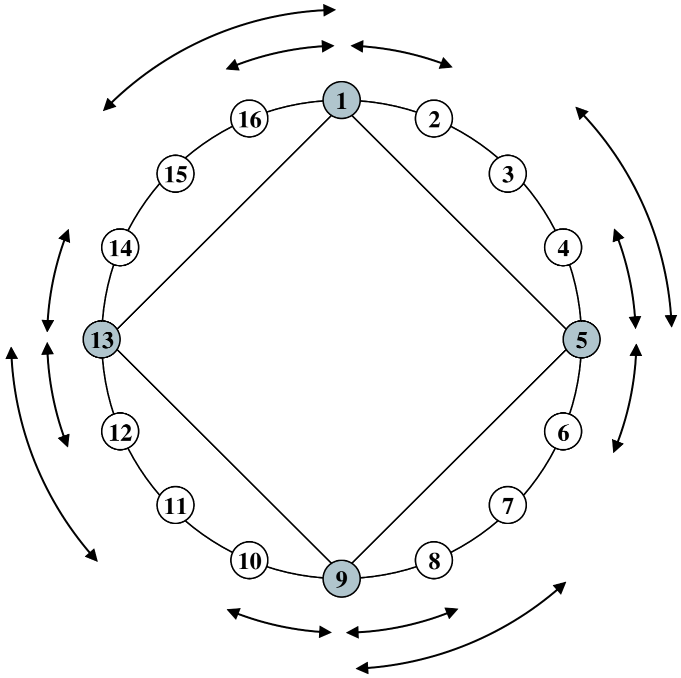


Figure 3-5: Assigning local wavelengths for a single-port 16-node ring.

acts as though it has $P' = P\lceil N/4 \rceil$ ports. From the results earlier in this section, we know that $P\lceil N/4 \rceil$ bypass wavelengths are necessary and sufficient to support such traffic, and furthermore that we can do so without the use of wavelength converters. (To see this, note that in the 4-hub ring $N = 4$ and $P' = P\lceil N/4 \rceil$.)

A summary of the properties of the ring banding algorithm follows:

Local Node Requirements: Each local node can use fixed routes and wavelengths to communicate with its nearest hub node. This allows the local node architecture to be extremely simple. In fact, using this algorithm, not only do local nodes never need to access bypass wavelengths, they do not need to switch local wavelengths either since these wavelengths are statically assigned. Therefore, the switching block shown in Figure 3-2 can be replaced with static connections from the P ports to the P assigned wavelengths.

Hub Node Requirements: Each of the four hub nodes needs to be able to switch between any of the local wavelengths and any of the bypass wavelengths. Bypass wavelengths never need to be switched onto other bypass wavelengths – i.e. no conversion is required between bypass wavelengths. $\lceil PN/2 \rceil$ converters between bypass and local wavelengths are required at each hub node.

Call rearrangements: Call departures never require existing calls to be rearranged since the wavelengths used can just be removed. It can be shown that call arrivals may require the rearrangement of at most 7 bypass wavelengths [31] (two sets of single-port traffic for the 4 hub nodes, less one for the newly arriving call). Local wavelengths never need rearrangement, as they are statically assigned.

We illustrate the operation of the 4-hub ring banding algorithm with an example.

Example 3. *Figure 3-5 considers single-port traffic on a 16-node ring, and shows the wavelengths that would be assigned in Steps 1 and 3 of the algorithm to handle the traffic between the local nodes and the hubs. Note that only the minimum number* $\lceil (16/4 - 1)/2 \rceil = 2$ *of local wavelengths are required. For any fixed traffic set, the four hubs would then provide a RWA using the bypass wavelengths via the* $\lceil PN/4 \rceil$ *algorithm.*

To illustrate the full RWA for a given call, consider a call travelling from node 6 to node 12. The closest hub for node 6 is 5, and the closest hub for node 12 is 13. Therefore a route is assigned as follows: (1) from node 6 to node 5 via a local wavelength, (2) from node 5 to node 13 via a bypass wavelength, and (3) from node 13 to node 12 via a local wavelength again. Figure 3-6 shows an example of a possible RWA for the call using this approach.

3.3 Torus Networks

Torus networks with no wavelength conversion were considered in [33], which presented a RWA algorithm based on column-first routing requiring twice the minimum number of wavelengths. We investigate the use of wavelength conversion to reduce the number of wavelengths required. We will first show a lower bound on wavelengths required; subsequently, we give a novel algorithm for RWA on a torus that makes efficient use of wavelengths and achieves the bound in certain cases.

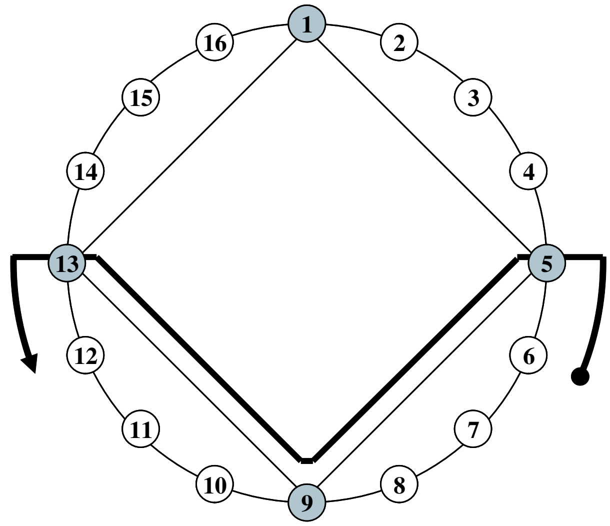


Figure 3-6: The route for call $(6, 12)$. Hubs 5 and 13 are used to access the bypass wavelengths.

3.3.1 Torus Lower Bound

Define an $R \times C$ torus to be a network consisting of RC nodes, each of which is uniquely assigned to a row r, $1 \leq r \leq R$, and a column c, $1 \leq c \leq C$. Each node is connected via four bidirectional links to four other nodes: the two adjacent nodes occupying the same row, and the two adjacent nodes in the same column. More specifically, denote the node in row r and column c by $n_{r,c}$. Then the node $n_{r,c}$ is connected via bidirectional links to the nodes $n_{r\oplus 1,c}$, $n_{r\ominus 1,c}$, $n_{r,c\oplus 1}$, and $n_{r,c\ominus 1}$, where operations on the rows and columns are modulo R and C, respectively. Figure 3-7 shows an example of a 6×3 torus. **Note that each row contains C nodes, and each column contains R nodes**.

We again use the P-port traffic model, where each node is assumed to have P ports, and hence can send and receive at most P calls. We next investigate how many wavelengths must be provisioned in the torus to support any admissible traffic set under these assumptions.

Suppose without loss of generality that $R \geq C$. (If the opposite is true, rotate the picture of the torus 90 degrees and re-label the rows as the columns and vice versa.) Let R be even. Consider a horizontal cut across the columns which removes

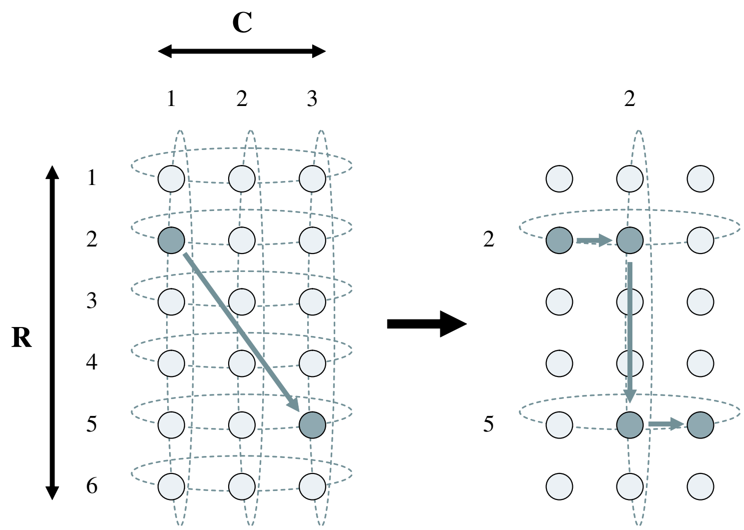


Figure 3-7: Breaking up a call into three sub-calls using a bridging column. The single call on the left, from $n_{2,1}$ to $n_{5,3}$, is routed as two row-ring calls and a column-ring call using the bridging column 2. Each of the sub-calls can be routed independently on their respective rings using the $\lceil PN/4 \rceil$ algorithm. Additional wavelength conversion may be required at nodes $n_{2,2}$ and $n_{5,2}$ if the sub-calls are not assigned to the same wavelength.

$2C$ links and divides the torus into two equal sets of $RC/2$ nodes each. Consider the first set. Under the P-port model, there exists a worst-case traffic set in which each node in that set sends all P calls to some node in the other set. In this case, there are $PRC/2$ calls traversing $2C$ links, which means that a minimum of $\lceil (PRC/2)/(2C) \rceil = \lceil PR/4 \rceil$ wavelengths are required. A similar argument for R odd yields a bound of $\lceil P(R-1)/4 \rceil$ wavelengths.

3.3.2 The TERA Algorithm - Overview

In this section we describe an algorithm based on the Torus Embedded-Ring Approach (TERA) for routing and wavelength assignment. We will show that by judicious use of wavelength conversion, the TERA algorithm will use no more than $\max\{\lceil PC/2 \rceil, \lceil PR/4 \rceil\}$ wavelengths. For toruses where $R \geq 2C$, this achieves the lower bound of $\lceil PR/4 \rceil$ wavelengths; in the worst case ($R = C$) it uses $\lceil PR/2 \rceil$.

We will describe the algorithm in detail later in this section, but the general idea is as follows. For any given call going from n_{r_1,c_1} to n_{r_2,c_2}, instead of considering all possible route assignments (of which there are many), we break the problem down into finding a route from the source n_{r_1,c_1} to some intermediate node n_{r_1,c_b} in the same row, from n_{r_1,c_b} to some n_{r_2,c_b} in the same column, and finally from n_{r_2,c_b} to the destination n_{r_2,c_2} in the same row. The advantage to this approach is that instead of having a single call travelling through a torus, the call has been subdivided into three smaller calls, each on a different ring, and the results for rings in Section 3.2 can be used to do the routing and wavelength assignment for each sub-call. Figure 3-7 gives an example of routing a call using this approach.

In the above approach, an intelligent choice of a column c_b for each call is required so that subsequent routing of the resulting sub-calls then uses as few wavelengths as possible.

For notational purposes, define the ring formed by the nodes $\{n_{i,j}|j = 1, ..., C\}$ to be the row-ring i, and the ring formed by the nodes $\{n_{i,j}|i = 1, ..., R\}$ to be the column-ring j. Under this nomenclature, there are R row-rings and C column-rings. Let us call the columns $\{c_b\}$ used to generate the sub-calls the *bridging columns*, since calls will use these columns to travel between row-rings. In an $R \times C$ torus, there are C bridging columns. We will use a bipartite matching approach to associate each call with a bridging column in such a way that the resulting sub-calls will form a $2P$-port traffic set on each row-ring, and a P-port traffic set on each column-ring. Once this is done, it is evident that by using the $\lceil PN/4 \rceil$-algorithm on each row and column-ring, the total number of wavelengths required in the torus will be $\max\{\lceil 2PC/4 \rceil, \lceil PR/4 \rceil\}$ $= \max\{\lceil PC/2 \rceil, \lceil PR/4 \rceil\}$, as claimed.

3.3.3 Bridging Column Assignment

In this section, we will describe a method for using matchings to assign a bridging column to each call such that the resulting sub-calls form a $2P$-port traffic set on each row-ring, and a P-port traffic set on each column-ring.

Consider a call $(n_{r_1,c_1}, n_{r_2,c_2})$ that is divided into three sub-calls $(n_{r_1,c_1}, n_{r_1,c_b})$, $(n_{r_1,c_b}, n_{r_2,c_b})$, and $(n_{r_2,c_b}, n_{r_2,c_2})$. We will call the $(n_{r_1,c_1}, n_{r_1,c_b})$ the *starting sub-call*,

$(n_{r_1,c_b}, n_{r_2,c_b})$ the *bridging sub-call*, and $(n_{r_2,c_b}, n_{r_2,c_2})$ the *ending sub-call*. We wish to determine the conditions that the the bridging column assignment are subject to.

1. Row-ring conditions: For each row-ring of size C, there are $2PC$ sub-calls to be routed on it, P of which are starting sub-calls and the remaining P of which are ending sub-calls. Each node already uses P ports for sending the starting sub-calls, and P ports for receiving the ending sub-calls. We wish to choose bridging columns such that the starting sub-calls on each row-ring use no more than P additional destination ports per node, and the ending sub-calls use no more than P additional source ports per node. If these conditions are met, then each row-ring needs no more than $2P$ ports per node.

2. Column-ring conditions: For each column-ring, we wish to choose bridging columns such that the P-port assumption holds true for each column-ring; i.e. the set of PR bridging calls for each column-ring should use no more than P source and destination ports per node.

Lemma 10. *If the bridging conditions imposed by the column-rings are satisfied, then the bridging conditions for the row-rings are also satisfied.*

Proof. For each row-ring, at least P ports are required regardless of the choice of column-rings since the sources of each starting sub-call and the destinations of each ending sub-call are fixed. Thus it remains to show only that each node is the destination of no more than P starting sub-calls, and the source of no more than P ending sub-calls, to prove that only another P ports per node are required (for a total of $2P$ per node).

To see that this is true, note that the destination of each starting sub-call is the source of a corresponding bridging sub-call. Therefore, if a node in a row-ring were the destination of more than P starting sub-calls, that node would also be the source of more than P bridging sub-calls on its column-ring. Since we assumed that the column-ring conditions were satisfied, this cannot be true.

Similarly, the source of each ending sub-call is the destination of a corresponding bridging sub-call. If a node in a row-ring were the source of more than P ending sub-calls, that node would also be the destination of more than P bridging sub-calls on its column-ring, which cannot be true. □

Lemma 10 tells us that it suffices to assign calls to bridging columns so that the column-ring conditions are satisfied. We achieve this by using a bipartite matching approach, as follows.

Consider the traffic set for the $R \times C$ torus. We construct a bipartite graph consisting of two sets of R vertices, which we will call $\{S_i\}$ and $\{D_j\}$, where $i = 1, \ldots, R$ and $j = 1, \ldots, R$. In a given set, each vertex corresponds uniquely to one of the R row-rings.

Each call in the traffic set will correspond to an edge in the graph. A call from a node n_{r_1,c_1} to n_{r_2,c_2} will be represented by an edge from vertex S_{r_1} to D_{r_2}; in other words, a call in the torus is represented by an edge in the graph connecting the vertex in $\{S_i\}$ representing its source row-ring, and the vertex in $\{D_j\}$ representing its destination row-ring. Since we have a P-port traffic set, and each row-ring contains C nodes, each vertex in the graph will have degree PC.

We will call the bipartite graph thus constructed the *bridging graph*. In the next step, we will make use of the following theorem for bipartite matchings with equal nodal degree.

Theorem 9. *Define a* perfect matching *to be a set of edges where exactly one edge is incident on every vertex. Then, in a bipartite graph $(V_1, V_2, \mathcal{E})$ in which each vertex in V_1 and in V_2 has degree m, the set $\mathcal{E}$ can be partitioned into m disjoint perfect matchings.*

Proof. The proof is basically by induction using Hall's theorem [37] and was given in [22]. It is omitted here for brevity. □

In the context of our constructed bipartite matching, Theorem 9 guarantees that we can obtain a set of PC disjoint perfect matchings. Each perfect matching corresponds to a set of R calls where exactly one call originates from each row-ring, and one call is destined for each row-ring. Since for any given call the bridging sub-call has source node equal to the source row of the original call, and destination node equal to the destination row of the original call, this means that if all calls in a matching use the same bridging column, *the set of resultant bridging sub-calls will correspond to a single-port traffic matrix for that column.*

The preceding idea forms the basis for the assignment of bridging columns. Recall that Theorem 9 guarantees that we will have PC disjoint perfect matchings. Divide these matchings into C sets of P disjoint perfect matchings. Assign each set of matchings to one of the C columns. All calls in a matching assigned to a given column will use that column as its bridging column. Since each matching requires only a single port per node, the P matchings in each column will require no more than P ports per node. Thus the column-ring conditions (and subsequently the row-ring conditions, by Lemma 10) are satisfied.

Example 4. *In this example, we consider the problem of assigning bridging columns to a traffic set on a single-port 4×2 torus. The traffic set is given in Figure 3-8. The corresponding bridging graph is shown in Figure 3-9. Recall that Theorem 9 states that since the graph has vertex degree $PC = 1 \cdot 2 = 2$, we can find 2 disjoint matchings. One possible such choice is given.*

Under the choice of matchings given in Figure 3-9, we assign all calls in the first matching to column 1, and all calls in the second matching to column 2. This uniquely specifies a bridging column for each call in the traffic set, as shown in Figure 3-10. For example, the first call in the set, from $n_{1,1}$ to $n_{3,2}$, corresponds to the graph edge S_1 to D_3. This edge is in the first matching, so the call from $n_{1,1}$ to $n_{3,2}$ is assigned to the bridging column of 1. The resulting sub-calls are ($n_{1,1}$,$n_{1,1}$), ($n_{1,1}$,$n_{3,1}$), and ($n_{3,1}$,$n_{3,2}$). (Note that the ($n_{1,1}$,$n_{1,1}$) sub-call happened to be degenerate, and in practice would not require a wavelength.)

3.3.4 The TERA Algorithm - Operation

We can now formally state the TERA algorithm.

THE TERA ALGORITHM

source		destination		graph edge	
row	column	row	column	S_i	D_j
1	1	3	2	1	3
1	2	2	1	1	2
2	1	1	1	2	1
2	2	4	1	2	4
3	1	2	2	3	2
3	2	1	2	3	1
4	1	4	2	4	4
4	2	3	1	4	3

Figure 3-8: A traffic set for the single-port 4x2 torus considered in Example 4. The first two pairs of columns give the row-column pairs for the source and destination nodes, while the last two columns give the edges that represent each respective call in the bridging graph.

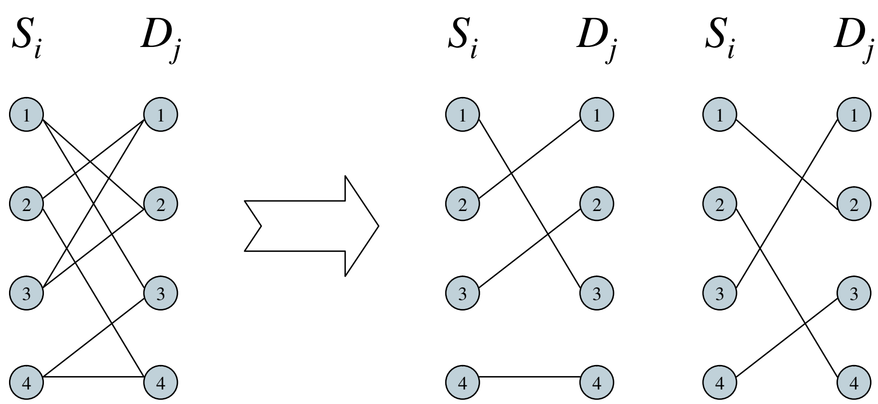


Figure 3-9: The bridging graph for Example 4. As expected, each vertex in the bridging graph has vertex degree $PC = 1 \cdot 2 = 2$. Using Theorem 9, this can be divided into $C = 2$ disjoint perfect matchings.

call		bridging column
src row	src col	
1	1	1
1	2	2
2	1	1
2	2	2
3	1	1
3	2	2
4	1	1
4	2	2

Figure 3-10: The resultant assignment of bridging columns to calls for Example 4

1. Given a P-port traffic set for the torus, construct the corresponding bipartite bridging graph, as described in Section 3.3.3.

2. Divide the edges on the bridging graph into C sets of P disjoint bipartite matchings. Assign each set to a different bridging column.

3. Now that each call has a bridging column, divide each call into a starting sub-call, a bridging sub-call, and an ending sub-call.

4. For each row-ring, use the $\lceil PN/4 \rceil$ algorithm to perform RWA on all starting and ending sub-calls within that row-ring. This requires $\lceil PC/2 \rceil$ wavelengths on each row-ring.

5. For each column-ring, use the $\lceil PN/4 \rceil$ algorithm to perform RWA of all bridging sub-calls within that column-ring. This requires $\lceil PR/4 \rceil$ wavelengths on each column-ring.

6. Give each original call in the torus the route and wavelength assignment formed by the concatenation of the RWA of the starting, bridging, and ending sub-calls. Up to two converters may be needed to change between sub-calls.

3.3.5 Banding and Bypass on Toruses

In this section, we give an algorithm for banding on a torus which reduces size of the band of local wavelengths dropped at non-hub nodes.

Since the TERA algorithm essentially results in a problem of disjointly routing traffic on different rings in the rows and columns, we can apply an approach similar to the ring banding algorithm. Rather than using the $\lceil PN/4 \rceil$ algorithm to route the sub-calls on the row-rings and column-rings, the ring banding algorithm is used. In this discussion, we focus on using the 4-hub ring banding algorithm, but the results could be extended to using different numbers of hubs.

We assume again by convention that $R \geq C$. Using a 4-hub architecture on each row, we can reduce the number of local wavelengths required on those rings to $P\lceil C/8 \rceil$. A minimum of $4R$ hubs are required to do this, since no two row-rings can share a hub node. In order for rings along the columns to also require a local band of no more than $P\lceil C/8 \rceil$ wavelengths, enough hubs must be allocated along the columns such that no node in a column-ring is further than $\lceil C/8 \rceil$ hops from a hub node. This requires at least $\lceil \frac{R}{\lceil C/4 \rceil} \rceil$ hub nodes along each column. An upper bound on the total number of hubs required is therefore $4R + \lceil \frac{R}{\lceil C/4 \rceil} \rceil \cdot C$, or approximately $8R$ hubs. Therefore the number of hubs h^* is bounded by $4R \leq h^* \leq 8R$. We can achieve the lower bound by using clever hub designation to allow the same nodes to serve as hubs for both a row-ring and a column-ring, reducing the total number of hubs required.

We describe a hub allocation scheme that uses the minimum number of hubs. For the first row, designate nodes $n_{1,h_1}, n_{1,h_2}, n_{1,h_3}$, and n_{1,h_4} to be hubs, where the column numbers h_i are given in Table 3.2. In the second row, nodes $n_{2,1\oplus h_1}, n_{2,1\oplus h_2}, n_{2,1\oplus h_3}$, and $n_{2,1\oplus h_4}$ are hubs; note that this is a cyclic shift in the right-ward direction (modulo C) of the hub allocation for the previous row. This pattern repeats for all subsequent rows. In general, $n_{r,c}$ is a hub if $c \oplus (r-1) = h_i$ for some $i = 1, 2, 3$, or 4; i is called the *hub index* of that hub.

It is trivial to note that each row now has 4 hubs, and hence only $P\lceil C/8 \rceil$ local

Table 3.2: Hub column numbers for the first row-ring

If C mod 4 = . . .	h_1	h_2	h_3	h_4
0	0	C/4	2 C/4	3 C/4
1	0	$\lfloor C/4 \rfloor$	$2\lfloor C/4 \rfloor$	$2\lfloor C/4 \rfloor + \lceil C/4 \rceil$
2	0	$\lceil C/4 \rceil$	$\lfloor C/4 \rfloor + \lceil C/4 \rceil$	$\lfloor C/4 \rfloor + 2\lceil C/4 \rceil$
3	0	$\lceil C/4 \rceil$	$2\lceil C/4 \rceil$	$3\lceil C/4 \rceil$

wavelengths are required along the rows. The following lemma claims that this hub allocation also requires no more than $P\lceil C/8 \rceil$ local wavelengths along the columns.

Lemma 11. *Designating hub nodes along the row-rings as described results in a hub allocation with the property that along each column-ring, no local node is more than $\lceil C/8 \rceil$ hops away from the nearest hub node.*

Proof. The proof will show that along any column, no two hubs are separated by more than $\lceil C/4 \rceil$ nodes, from which it follows that no node can be more than $\lceil C/8 \rceil$ hops away from a hub.

Recall that a node $n_{r,c}$ is a hub if $c \oplus (r-1) = h_i$; equivalently, $n_{r,c}$ is a hub iff $r = h_i \ominus (c-1)$ for some $i = 1, 2, 3,$ or 4. Consider two adjacent hubs in the same column $n_{r_1,c}, n_{r_2,c}$ with hub indices i and j; the distance between them is $r_1 - r_2 = h_i - h_j$. From Table 3.2, for any two adjacent h_i and h_j, $h_i - h_j \leq \lceil C/4 \rceil$, completing the proof. □

The consequence of the lemma is that any local node is no more than $\lceil C/8 \rceil$ nodes away from a hub; hence no more than $P\lceil C/8 \rceil$ wavelengths are required. Figure 3-11 gives an example of a 21×21 torus, and illustrates the hub allocation obtained from this construction.

The number of bypass wavelengths required can be obtained by examining the number of calls arriving and departing from each hub node. Along the rows, each hub is responsible for $\lceil C/4 \rceil$ nodes, and each such node has $2P$ ports; therefore each row-hub has $P' = 2P\lceil C/4 \rceil$. Using the $\lceil PN/4 \rceil$ algorithm, $P\lceil C/2 \rceil$ bypass

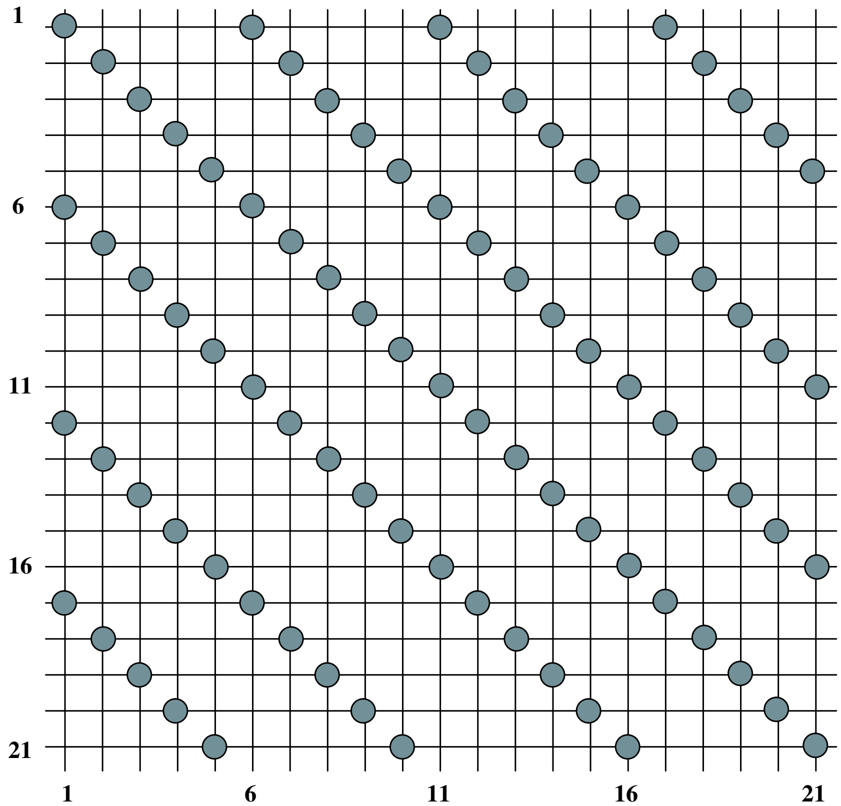


Figure 3-11: A 21×21 torus. The local nodes are at all intersection points of the grid, while the hub nodes are shown as shaded circles. For $R = C = 9$, we have that $\lfloor N/4 \rfloor = 2$ and $\lceil N/4 \rceil = 3$, so that $h_1 = 0$, $h_2 = 2$, $h_3 = 4$, and $h_4 = 7$. Adding each of the row numbers modulo 9 gives the hub assignments shown. As a check, note that $g_1 = 0$, $g_2 = 2$, $g_3 = 5$, and $g_4 = 7$ also correctly yields the resulting hub allocations down the columns.

wavelengths are required. A similar argument on the columns shows that $P\lceil R/4 \rceil$ bypass wavelengths are required.

The wavelength assignment also has the following properties:

1. *Non-hub nodes have fixed routing and fixed wavelengths.* Conversion between local wavelengths is hence not required, since each node is assigned its own wavelength to send and receive from its hub. Also, local nodes need no knowledge of network state.

2. *The only place where conversion is required is at the hubs.* No conversion is required to connect calls continuing from a row-ring onto a column-ring, or from a column-ring onto a row-ring.

3.4 Tree Networks

3.4.1 Tree Lower Bound

In this section, a bound on the minimum number of wavelengths required to support P-port traffic in trees is established. We use the cut-set bound to obtain a lower bound on the number of wavelengths. Since a tree contains no cycles, the removal of any single link i disconnects the tree into two disjoint sets of nodes. For each i, we call these sets S_i^1 and S_i^2, and let $|S_i^1|$ and $|S_i^2|$ denote the number of nodes in each set, respectively. Suppose $|S_i^1| \leq |S_i^2|$. Then there exists a worst-case admissible set where each node in S_i^1 sends all P units of traffic to some node in S_i^2. Since all this traffic must cross link i, at least $W_i = |S_i^1|$ wavelengths are required to support it. If $|S_i^1| > |S_i^2|$, similar reasoning gives $W_i = |S_i^2|$.

We can obtain the tightest lower bound by maximizing over all links i. Let the greatest lower bound thus obtained be $W = \max_i\{W_i\} = \max_i\{\min\{|S_i^1|, |S_i^2|\}\}$. A link which achieves this lower bound is known as a *bottleneck link*. Figure 3-12 gives an example of determining the bottleneck link.

From the preceding discussion, it is clear that the lower bound obtained in this section is dependent on the specific topology of the tree, and not just on the number

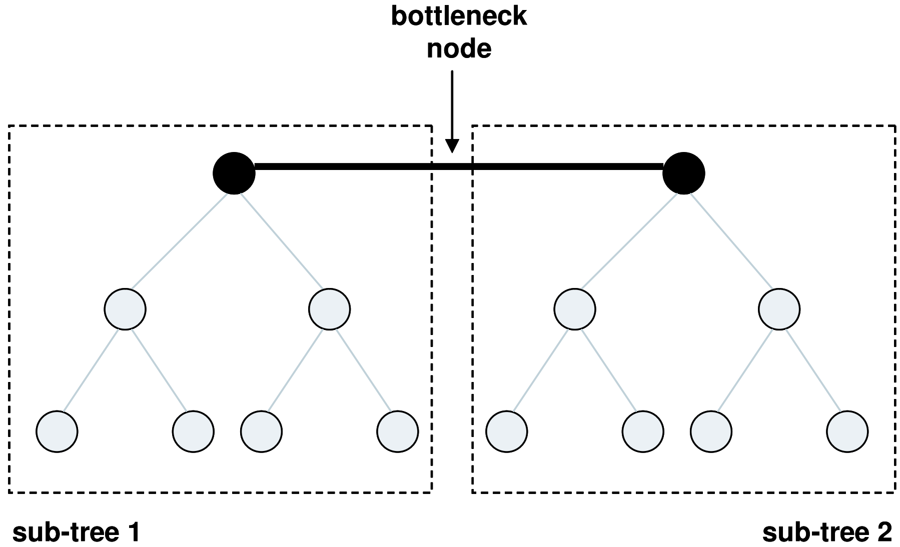


Figure 3-12: A 14-node balanced binary tree. The bottleneck link is the link in heavy black; removal of this link from the graph disconnects the graph into the two indicated sub-trees, each containing $|S_i^1| = |S_i^2| = 7$ nodes. By considering all links in turn, it can be shown that this is the bottleneck link, resulting in a lower bound of 7 wavelengths.

of nodes in the tree. For example, for a balanced d-ary tree, the links adjacent to the root node are the bottleneck links, and $W = P(N-1)/d$. For balanced binary trees, where $d = 2$, the bound is $W = P(N-1)/2$.

3.4.2 The $\lceil PN/2 \rceil$ Embedded-Ring Approach

In this section, we describe a RWA based on embedding a virtual ring in the tree topology. We will show that for connected P-port traffic sets, this approach requires at most $\lceil PN/2 \rceil$ wavelengths for any tree topology, and hence is optimal for tree topologies where $W = \lceil PN/2 \rceil$. For example, it is optimal for balanced binary trees. Furthermore, no wavelength conversion is required.

The ring-embedding idea is intuitively very simple. In any tree, by using depth-first search, we can form a circuit which visits each node in the tree at least once while traversing each link only twice (once in each direction). This circuit is said to form a *virtual ring* in the following sense. Consider a ring topology where the nodes are connected in the order in which each corresponding node in the tree is first visited by the circuit. Then any RWA for this ring has a one-to-one correspondence with a RWA for the tree. Each link between two adjacent nodes on the ring corresponds to the links traversed by the circuit in travelling between those two nodes on the tree. Such a circuit is illustrated in Figure 3-13 for a 15-node balanced binary tree.

A single, unidirectional wavelength on the ring corresponds to the use of a single, bidirectional wavelength on the tree. (A bidirectional wavelength is required because a single circuit around the tree used each link once in each direction.) Recall from Lemma 9 that any two adjacent calls on a ring must fit on a single wavelength in one direction or the other. The ring embedding algorithm simply divides the traffic into adjacent pairs, and determines the single-wavelength direction for each pair. Each pair is then routed on a single directed wavelength on the virtual ring, which corresponds to a RWA on the tree which uses a single bidirectional wavelength per pair. Since there are a total of $\lceil PN/2 \rceil$ pairs, no more than $\lceil PN/2 \rceil$ wavelengths are required.

3.4.3 Banding and Bypass using the Embedded-Ring Approach

Banding can also be performed using the embedded-ring approach by directly applying the ring banding algorithm on the virtual ring. For a 4-hub architecture, for example, the approach of Section 3.2.4 can be directly used to obtain a RWA for the virtual ring which uses $\lceil PN/8 \rceil$ local wavelengths and $\lceil PN/4 \rceil$ bypass wavelengths. Since a single unidirectional wavelength on the virtual ring requires a bidirectional wavelength on the tree, this implies that $2\lceil PN/8 \rceil$ local wavelengths and $2\lceil PN/4 \rceil$ bypass wavelengths are required for the tree using this approach.

3.5 Chapter Summary

We considered routing and wavelength assignment for P-port traffic on ring, torus, and tree topologies with and without wavelength conversion. For each topology, we

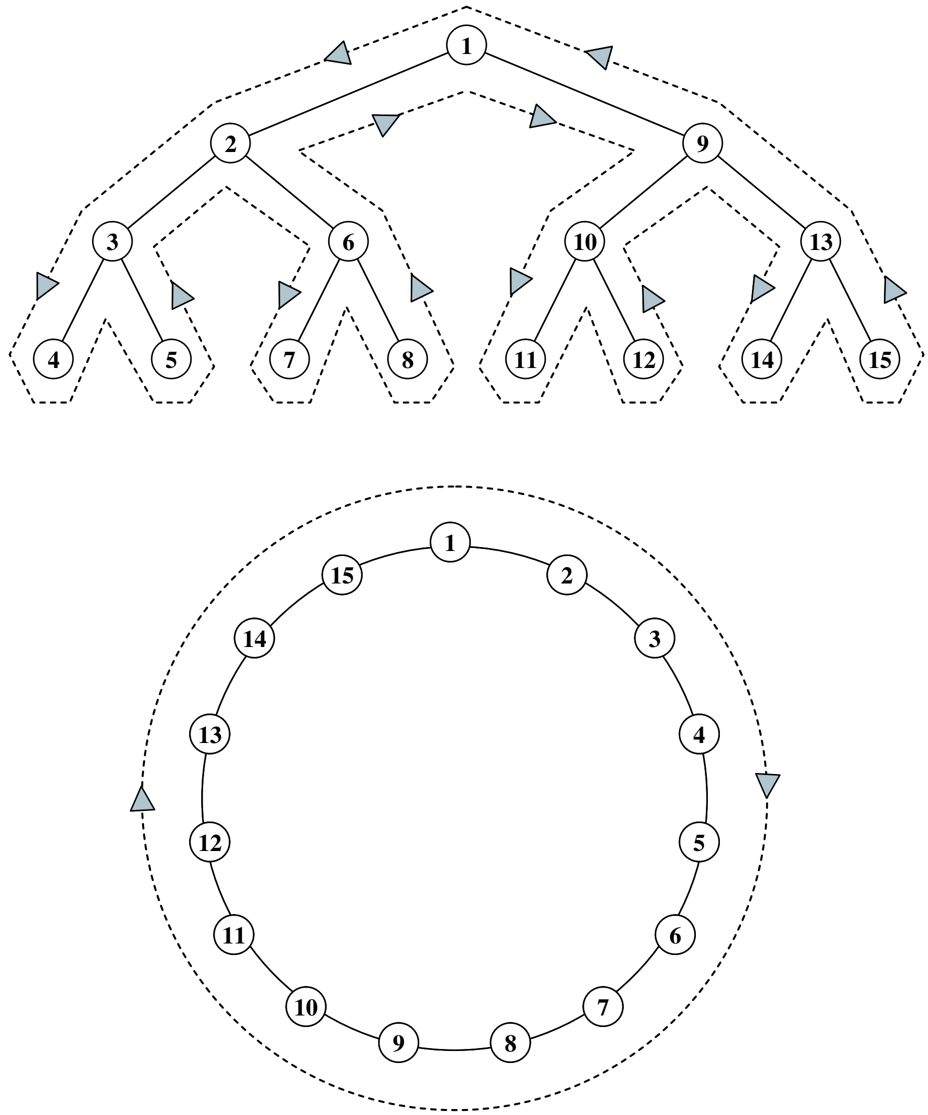


Figure 3-13: Embedding a cycle in a 15-node balanced binary tree. The nodes have been numbered so that the cycle visits them in order of increasing index. The corresponding virtual ring topology is shown below the tree.

first used the objective of minimizing wavelengths, then moved on to maximizing optical bypass via banding.

For rings, we recap the $\lceil PN/3 \rceil$ and $\lceil PN/4 \rceil$ algorithms from [24] and [6] which are optimal in minimizing the number of wavelengths used for networks without and with wavelength conversion, respectively. We then developed a novel approach for banding which minimizes the size of the band of local wavelengths dropped at all nodes, and additionally eliminates the need for switching at all but h hub nodes, where h is a design parameter. Furthermore, only the hubs need to have knowledge of the network state, eliminating the need for a centralized management scheme which needs to coordinate with all nodes.

For toruses, we derive a lower bound of $\lceil PR/4 \rceil$ wavelengths required to support P-port traffic, and provide a novel algorithm (the TERA algorithm) which uses ring embeddings and requires $\max\{\lceil PC/2 \rceil, \lceil PR/4 \rceil\}$ wavelengths for an $R \times C$ torus using sparse conversion. We show that this is optimal for $R \geq 2C$. This improves over the algorithm in [33] which requires $\lceil PR/2 \rceil$ wavelengths. We then provide a banding algorithm based on the ring banding approach which eliminates switching at all but $4R$ hubs. Again, only the hubs need information about the network state in this second RWA.

Finally, for trees, we describe a method based on the cut-set bound which obtains a lower bound on the number of wavelengths required to support any P-port tree. We then use a ring embedding approach to produce the $\lceil PN/2 \rceil$ algorithm, which has low complexity and is optimally efficient in wavelength usage for a certain subset of trees, including balanced binary trees. A method for applying the ring banding algorithm to trees for bypass requiring only half the wavelengths to be dropped at local nodes is then derived.

Chapter 4

Multigranularity Switching

We have already discussed in Chapters 2 and 3 two approaches to network architecture design. First, we considered fully-switched systems and attempted to minimize the total number of wavelengths in the system. We then investigated reducing nodal complexity by allowing only partial wavelength accessibility at each node, allowing the majority of the wavelengths to cheaply bypass the node.

In this chapter, we propose a second option for reducing hardware costs. Thus far, we have considered only wavelength-level switches. In any reasonably-sized network, the number of wavelengths on the fiber is typically much greater than the nodal degree. The result is that many wavelengths in each input fiber at any given node will be switched to the same output fiber. Rather than switching each wavelength individually, *waveband switches* can be used to switch all the wavelengths using a single port, instead of requiring a separate port for each wavelength.

Since switch costs rise very quickly with the number of ports required, reducing the number of switch ports can have significant cost impact. In this chapter, we investigate the use of multigranularity switching to reduce switch costs. We will consider the number of switch ports and the number of wavelengths to be the figures of interest, and attempt to develop an understanding of the fundamental tradeoffs between these resources that can be achieved.

4.1 System Model

In this chapter, we adopt the P-port traffic model from [24], which assumes that P transmitters and receivers are available at each network node. This allows each node to send and receive a total of at most P calls at any given time. If the instantaneous traffic is represented by a matrix where each entry (i, j) consists of the number of calls sent from node i to node j, the P-port model constrains each row and column sum to be at most P. Any traffic set with a matrix obeying this constraint is termed *admissible*, and no calls in an admissible set may be blocked. Under this model, sufficient resources must be provisioned to support any admissible set. Call arrivals and departures may occur in arbitrary fashion, as long as the resultant traffic set remains admissible; these dynamic arrivals and departures are represented by transitions between different admissible sets. This model is attractive because it limits traffic in a realistic fashion based on hardware constraints, and also allows dynamic aspects of the traffic to be captured without making assumptions about the call statistics.

We primarily consider the star topology in this chapter, with Section 4.5 describing extensions of our results to other topologies. This topology is representative of a hub or switch node in a network. All nodes are connected via bidirectional fibers to a central hub, which performs the switching. We assume no wavelength conversion, so calls must use the same wavelength on all hops. To avoid collision, no two calls may use the same wavelength in the same direction on the same fiber. We consider both the single-source case, where only a single node transmits, and the more general multi-source case, where every node may transmit. These situations are illustrated in Figure 4-1.

The problem of band switching under this model may be formulated as a matrix decomposition. Under the banding problem, we are given a traffic matrix C where each entry $[C]_{i,j}$ represents the number of calls transmitted from source node i to destination node j. In the single-source case, the traffic matrix is a vector of size $1 \times N$; for multi-source traffic, the traffic matrix is a square $N \times N$ matrix. Note that C may change over time due to call arrivals and departures. For a fixed C, the goal

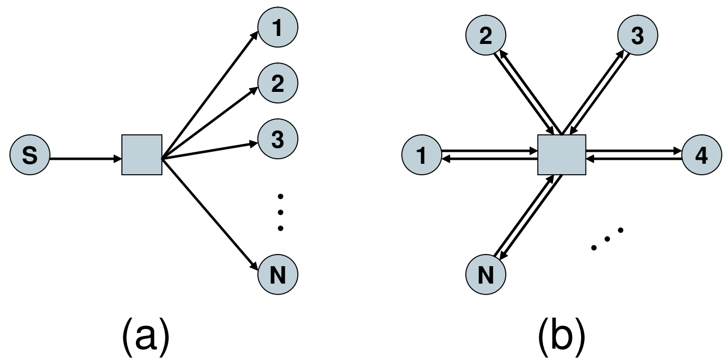


Figure 4-1: (a) Single-source case, where a single source node sends a total of at most P calls to up to N destinations. (b) Multi-source case, where each of N nodes sends and receives a total of P calls.

is to group the calls into bands such that calls within the same band that have the same source node go to the same destination. This can be expressed mathematically by

$$C \leq b_1 T_1 + b_2 T_2 + \ldots + b_B T_B \tag{4.1}$$

where each b_i is an integer representing the size of waveband i, and each T_i represents the corresponding switch configuration. T_i is therefore either a unit vector (single-source case) or a permutation matrix (multi-source).

Any set of wavebands specifying a valid decomposition is sufficient to support the particular traffic matrix C to which it applies. We impose two additional constraints: we require that B and $\{b_i\}$ must be fixed over all admissible traffic sets. Fixing B is essential since B corresponds to the number of switches required, a hardware requirement that should not depend on random changes in the traffic. Fixing the band sizes $\{b_i\}$ removes the need for dynamically tunable filters, reducing costs. Under these two constraints, we require that banding algorithms be characterized by fixed values of B and $\{b_i\}$ such that for each admissible traffic set C, the algorithm is able to specify a decomposition according to (4.1) with specific switch configurations T_i for each waveband i. Recall that the performance of each banding algorithm can be judged by the number of wavelengths and wavebands it requires. The number of wavebands is given directly by B, while the total number of wavelengths can be calculated as $\sum_{i=1}^{B} b_i$.

As illustrated in Figure 1-3, our goal is to find the optimal achievable frontier. The most general formulation of this problem is to allow the waveband sizes $\{b_i\}$ to be non-uniform. Any uniform-waveband algorithm will then be a valid special case. Under the general formulation, we can divide this problem into three parts. Note that Figure 1-3 shows two asymptotes to the achievable region: one corresponding to the minimum possible number of wavelengths required (the *minimum-wavelength asymptote*, shown on the bottom of the achievable region), and the other corresponding to the minimum number of wavebands (the *minimum-waveband asymptote*, shown to the left of the achievable region). The first two parts of the problem focus on these asymptotes, and attempt to determine the optimal points on these lines. Specifically, we denote any algorithm with performance achieving the minimum possible number of wavelengths to be a *minimum-wavelength algorithm*. The problem of finding the best minimum-wavelength algorithm is known as the *minimum-wavelength problem*. Similarly, any algorithm using the minimum possible number of wavebands is a *minimum-waveband algorithm*; finding the best minimum-waveband algorithm is the *minimum-waveband problem*.

Once solutions to the minimum-wavelength and minimum-waveband problems are obtained, two points on the optimal frontier are known. It then remains only to find the best tradeoff between wavelengths and wavebands achievable between these two points. Ideally, such a tradeoff should present a curve which adheres as closely as possible to the asymptotes, presenting the best possible tradeoff. Obtaining the best possible such tradeoff is the subject of the final component of the banding problem.

In Section 4.2, the banding problem is investigated for the single-source traffic case. In Sections 4.3 and 4.4, banding for general multi-source traffic is considered. We will compare the special case of uniformly-sized wavebands to the more general non-uniform case, and show that uniform waveband sizing compares very favorably. Finally, Section 4.5 describes extending the results of the chapter to general network topologies.

4.2 Waveband Switching for Single-Source Traffic

In this section, we consider the banding problem for the case of single-source traffic. Under this scenario, a single source node sends up to P units of traffic to be switched to N possible destination nodes. The primary purpose of investigating the single-source model is to derive intuition for the design of good banding algorithms that will be beneficial in addressing the multi-source traffic case in Section 4.3. The single-source model also has some merit in cases of a one-to-many traffic scenario.

Example 5. *Consider the case of $P = 4$, $N = 2$. In this example, the source sends 4 calls, distributed among 2 destinations. There are only 5 possible maximal traffic sets, which (expressed in vector form) are $[4,0],[3,1],[2,2],[1,3],$ and $[0,4]$. Clearly at least 4 wavelengths are required to support the traffic, since there are 4 calls. We can show that if we restrict ourselves to using only 4 wavelengths (i.e. we consider the minimum-wavelength problem), the minimum number of wavebands required is 3: one band of size 2, and two bands of size 1. By exhaustive verification we can prove that this waveband sizing is sufficient for all possible traffic sets:*

$$\begin{aligned}
[4,0] &= 2\cdot[1,0] + 1\cdot[1,0] + 1\cdot[1,0] \\
&= 2e_1 + e_1 + e_1 \\
[3,1] &= 2e_1 + e_1 + e_2 \\
[2,2] &= 2e_1 + e_2 + e_2
\end{aligned}$$

where e_i is a unit vector with the i^{th} entry equal to 1.

Note that, as required, the sizes of each band and total number of bands are fixed, and only the accompanying unit vectors (which correspond to the switch configurations for each band) change between traffic sets. The switching of each waveband for each scenario is illustrated in Figure 4-2. In this example, the savings in switching

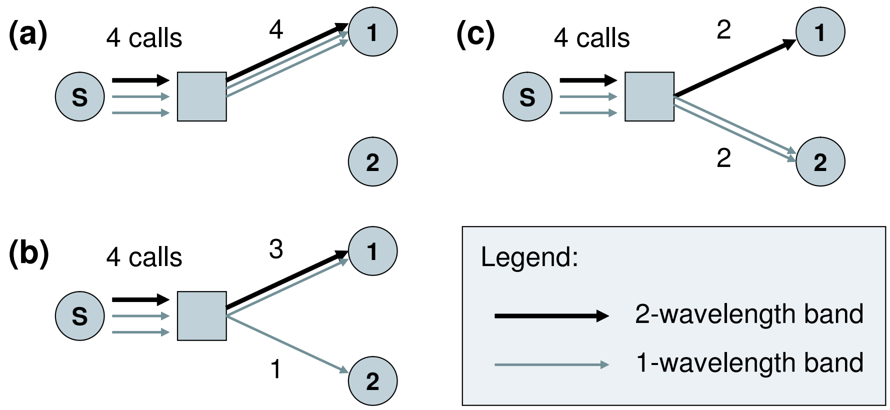


Figure 4-2: The switching configurations for the 3 unique maximal traffic sets of Example 5. The traffic sets shown are: (a) [4,0] (b) [3,1] (c) [2,2].

is not large because the number of calls is not very large relative to the number of destinations. As the number of calls increases, the savings will increase as well.

The number of wavebands can be further reduced if the use of additional wavelengths is permitted. One possibility is to have one band of size 3, and one band of size 2. This reduces the number of wavebands to two, and these two wavebands can still support all 5 possible traffic sets. However, the total number of wavelengths used has increased to 5. Efficient methods of making these sorts of tradeoffs will be discussed.

We consider the two special cases of the minimum-wavelength and minimum-waveband problems. Recall that the solutions to these two problems will provide two points on the optimal achievable performance frontier. We will defer the discussion of obtaining good tradeoffs between these two points until the multi-source case.

4.2.1 The Minimum-Wavelength Problem

Recall that for the minimum-wavelength problem, we consider only banding algorithms that use the minimum possible number of wavelengths. Under the P-port model, a minimum of P wavelengths are clearly necessary and sufficient: up to P calls can be sent, and if each wavelength is individually switched, P wavelengths can support all the calls. The goal is to find the optimal algorithm that uses only P

wavelengths.

Since the number of wavelengths equals the maximum possible number of calls, given a maximal admissible traffic set, each wavelength must be used to support a call. Therefore, the minimum-wavelength problem is equivalent to finding a method of partitioning the P wavelengths into wavebands such that, for any admissible traffic set, there exists a method for assigning calls to wavebands such that every wavelength is assigned a call. Furthermore, the optimal minimum-wavelength algorithm should accomplish this while at the same time minimizing the total number of wavebands. We show in this section that the optimal minimum-wavelength banding algorithm is a greedy algorithm. Specifically, the greedy algorithm chooses waveband sizes recursively, where at each step a waveband is chosen to be as large as possible subject to the constraint that every wavelength in that band can always be assigned a call under any maximal admissible traffic set. (We say that in this case every wavelength can be *fully utilized.*)

Define $b_{max}(N, P)$ to be the maximum waveband size that we can guarantee will be fully utilized by any traffic set sending P calls to N destinations. Since all calls in the waveband must go to the same destination, this is equivalent to providing a guarantee that a destination node can always be found (under *any* admissible P-port traffic set) which receives at least $b_{max}(N, P)$ calls. To illustrate, when considering the example in Figure 4-2, we note that over all admissible traffic sets, a destination can always be found which receives at least 2 calls, leading to the conclusion that $b_{max}(N, P) = 2$ in that case. In general, we can guarantee that at least one of the destinations receives $\lceil P/N \rceil$ calls. Furthermore, this is the largest number for which we can make this guarantee; this follows from the fact that one admissible traffic set is where the traffic is divided evenly (up to a difference of one wavelength due to integer constraints) among all destinations, and no destination receives more than $\lceil P/N \rceil$ calls under this traffic set. Therefore $b_{max}(N, P) = \lceil P/N \rceil$.

Single-Source Greedy Algorithm:

1. Let $P_1 = P$ be the number of calls remaining and N be the number of nodes. Let $i = 1$.

2. Let waveband i be of size $b_i = b_{max}(N, P_i) = \lceil P_i/N \rceil$.

3. Locate a destination receiving at least $b_{max}(N, P_i)$ calls. Route waveband i to this destination, and assign $b_{max}(N, P_i)$ calls to it. The number of calls remaining becomes $P_{i+1} = P_i - b_{max}$.

4. If $P_{i+1} > 0$, let $i \leftarrow i + 1$ and go to Step 2.

Example 6. *We revisit Example 5 and show how the greedy algorithm is used to obtain the minimum-wavelength waveband sizes used in Figure 4-2. In that example,* $P = P_1 = 4$ *and* $N = 2$*. In the first iteration, the greedy algorithm chooses the first waveband to be of size* $\lceil P_1/N \rceil = \lceil 4/2 \rceil = 2$*. As a corollary, we are guaranteed that 2 calls can be assigned to this waveband, leaving* $P_2 = 2$ *calls unassigned. The next two iterations partition the remaining wavelengths into bands of a single wavelength each, for a final partition of* $\{2, 1, 1\}$*.*

We must now show that choosing the waveband sizes using the greedy algorithm is optimal for the minimum-wavelength problem. The full proof, omitted here for brevity, is based on establishing that the minimum number of wavebands required to support a given number of calls is non-decreasing in the number of calls. It therefore follows that the optimal approach for choosing each waveband size is to choose the band as large as possible, thereby minimizing the amount of traffic which remains (and therefore the subsequent number of wavebands).

The greedy algorithm provides a method for optimally determining waveband sizes for the minimum-wavelength problem. This also implicitly provides a way of determining the minimum number of wavebands required (i.e. by running the algorithm and counting the number of wavebands produced). We can also derive an explicit upper bound on the minimum number of wavebands required in the minimum-wavelength scenario. We proceed by relaxing the integer constraints on $b_{max}(N, P)$. Let P_k be the number of calls remaining after running the k^{th} iteration of the greedy algorithm. The series progresses as follows:

$$\begin{aligned}
P_1 &= P - \frac{P}{N} = \left(1 - \frac{1}{N}\right) \cdot P \\
P_2 &= \left(1 - \frac{1}{N}\right) \cdot P_1 = \left(1 - \frac{1}{N}\right)^2 \cdot P \\
&\vdots \\
P_k &= \left(1 - \frac{1}{N}\right)^k \cdot P
\end{aligned} \tag{4.2}$$

If $P \leq N$, then the number of bands B is simply equal to P since each band is composed of only a single wavelength. Therefore consider $P > N$ and determine the number of bands k required to reduce the number of unassigned wavelengths to N.

$$\begin{aligned}
P_k &= N \\
\left(1 - \frac{1}{N}\right)^k \cdot P &= N \\
k &= \frac{\log\left(\frac{N}{P}\right)}{\log\left(1 - \frac{1}{N}\right)}
\end{aligned}$$

Then the total number of wavebands is simply $k + N$. Since relaxing the ceiling constraints underestimates the size of each waveband, this gives an upper bound on the number of wavebands B, namely:

$$B \leq \begin{cases} N + \frac{\log\left(\frac{N}{P}\right)}{\log\left(1 - \frac{1}{N}\right)} & , \ P > N \\ \\ P & , \ P \leq N \end{cases} \tag{4.3}$$

From (4.3), we can also make the additional observation that if $P \leq N$, the number of bands B equals the number of wavelengths P, and there is no savings from banding in the minimum-wavelength case as each wavelength continues to be switched individually.

4.2.2 The Minimum-Waveband Problem

The optimal minimum-waveband algorithm is the one that requires the fewest wavelengths subject to using only the minimum number of wavebands. In addition to providing a second point on the optimal frontier, this will establish the minimum cost in wavelengths required to obtain the maximum possible reduction in switching.

If $P < N$, then there exists an admissible traffic set where each call is sent to a different destination, and P wavebands are necessary. It follows from a little further thought that P wavebands of a single wavelength each are actually also sufficient to support all admissible traffic sets, since each call can be assigned a dedicated waveband under this provisioning.

The more interesting case arises when $P \geq N$. Since all wavelengths in the same waveband must be switched to the same destination, and there are N possible destination, a minimum of N wavebands are are necessary. One (inefficient) approach that requires only N wavebands is to statically switch one waveband to each destination, and provision P wavelengths per waveband; since there are a total of only P calls, this is sufficient to support any admissible traffic set. Our goal is to find a better, optimal algorithm using only N wavebands that minimizes the number of wavelengths used. We first obtain a lower bound on the number of wavelengths required using the following lemma.

Lemma 12. *Consider a banding algorithm that uses N wavebands, and order the wavebands from smallest to largest. Let b_i be the size of the i^{th} waveband. If the source sends up to P calls to the N destination nodes, b_i is bounded by*

$$b_i \geq \left\lfloor \frac{P - N + i}{i} \right\rfloor, \quad i = 1, \ldots, N \tag{4.4}$$

Corollary: The total number of wavelengths W required is bounded by the sum of the bounds on the individual waveband sizes, namely

$$W \geq \sum_{i=1}^{N} \left\lfloor \frac{P-N+i}{i} \right\rfloor \tag{4.5}$$

This summation can be shown to increase as $O(P \log N)$.

Proof. The proof proceeds by constructing an admissible traffic set which requires b_i to have at least $\lfloor \frac{P-N+i}{i} \rfloor$ wavelengths. Consider the traffic set where the source S sends $\lfloor \frac{P-N+i}{i} \rfloor$ calls each to the first i nodes, and a single call to each remaining node. The total traffic in this construction is

$$i \cdot \left\lfloor \frac{P-N+i}{i} \right\rfloor + (N-i) \leq P$$

and therefore it is admissible. Since each destination receives at least one call, each of the N wavebands goes to a different destination.

Without loss of generality, we assign the largest i wavebands to nodes 1 through i. Each of these wavebands must support $\lfloor \frac{P-N+i}{i} \rfloor$ calls. Therefore $b_i \geq \lfloor \frac{P-N+i}{i} \rfloor$. □

Since Lemma 12 is a lower bound, any minimum-waveband algorithm which achieves the bound is optimal. We next present an algorithm that can support any admissible traffic set using wavebands of the minimal sizes specified by (4.4). Since this minimum-waveband algorithm would use no more wavelengths than the lower bound, it is therefore optimal.

<u>Min-Band Algorithm</u>:

1. Index the N waveband in order of decreasing size, so that waveband i has size $b_i = \lfloor \frac{P-N+i}{i} \rfloor$, where $i = 1, \ldots, N$. Note that b_1 is the largest waveband, and b_N is the smallest.

2. Let $i = 1$.

3. Locate the destination node with the greatest number of remaining calls. Switch waveband i to that node. Assign up to b_i calls to waveband i, and remove these calls from the traffic set.

4. If no calls remain, the algorithm terminates. Otherwise, increment i and return to Step 2.

By design, the algorithm uses only wavebands of the minimum size, and therefore meets the lower bound. It remains only to show that it is able to support any admissible set. First, suppose that each destination receives at least a single call. In this case, we can rank each destination in decreasing order of number of calls received, so that the first destination receives the most calls. Then the min-band algorithm allocates the i^{th} waveband to the i^{th} destination. Since each destination receives at least one call, the first i destinations receive at most $P - (N - i) = P - N + i$ calls, and the i^{th} destination receives at most $\lfloor (P - N + i)/i \rfloor$ calls. Since the i^{th} waveband has size $\lfloor (P - N + i)/i \rfloor$, it suffices to accommodate the calls. The proof in the case where some destinations do not receive calls is more cumbersome but follows the same approach.

Example 7. *Consider the case where $P = 22$ calls are distributed among $N = 4$ destinations. The optimum minimum-waveband algorithm requires 4 wavebands. According to Lemma 12, the first waveband is of size $b_1 = \lfloor P - N + 1 \rfloor = \lfloor 22 - 4 + 1 \rfloor = 19$. Similarly, $b_2 = 10$, $b_3 = 7$, and $b_4 = 5$.*

These wavebands can support any P-port admissible traffic set, $P = 22$. For example, consider the traffic set $C = [5, 8, 7, 2]$. The first waveband is assigned to node 2, the destination with the most traffic, and carries all 8 calls. Similarly, b_2 is assigned to node 3, b_3 is assigned to node 1, and b_4 is assigned to node 4. Using the matrix decomposition notation of (4.1), this can be written as

$$\begin{aligned} [5, 8, 7, 2] &\leq 19e_2 + 10e_3 + 7e_1 + 5e_4 \\ &= [7, 19, 10, 5] \end{aligned}$$

Note that here, the total number of wavelengths available to each destination (represented by the vector on the right-hand side of the equation) is greater than the number of calls: more wavelengths were provisioned than absolutely required for this particular traffic set. This over-provisioning in wavelengths is necessary in order to guarantee that all *admissible traffic sets can be accommodated.*

It is instructive to compare this to the decomposition obtained by the greedy algorithm of Section 4.2.1. The first waveband for the greedy algorithm consists of $\lceil P/N \rceil = \lceil 22/4 \rceil = 6$ *wavelengths. The remaining waveband sizes can be shown to be* $\{4, 3, 3, 2, 1, 1, 1, 1\}$*. One possible switching configuration for these waveband sizes is:*

$$\begin{aligned} [5, 8, 7, 2] &= 6e_2 + 4e_3 + 3e_1 + 3e_3 + 2e_1 + e_2 \\ &\quad + e_2 + e_4 + e_4 \\ &= [5, 8, 7, 2] \end{aligned}$$

The greedy algorithm, since it is a minimum-wavelength algorithm, did not over-provision any wavelengths; however, more wavebands were required. This example also illustrates an important point. In the decomposition given by (4.1), equality is guaranteed to hold for all C *if and only if the wavebands were allocated using a minimum-wavelength algorithm (such as the greedy algorithm). Algorithms such as the min-band algorithm allow some overprovisioning of wavelengths in order to further decrease the number of wavebands required.*

Inequality (4.5) gives total the number of wavelengths required by the optimal minimum-waveband algorithm. We can approximate the total number of wavelengths in closed form by relaxing the integrality constraint on the terms of the summation. This gives:

$$
\begin{aligned}
W_{min} &\approx \sum_{i=1}^{N} \frac{P-N+i}{i} \\
&= N + (P-N)\sum_{i=1}^{N}\frac{1}{i} \\
&\approx N + (P-N)\log N \\
&\approx P\log N
\end{aligned}
$$

where the last approximation holds for $P \gg N$.

Recall that any banding algorithm always requires a minimum of P wavelengths. Our result in this section shows that the best minimum-waveband algorithm requires $O(P\log(N))$ wavelengths. One way of interpreting this result is that $\log(N)$ represents the minimum *wavelength inefficiency* necessary to achieve the minimum-waveband bound. In other words, if we desire to use no more than the minimum number of wavebands N, then we must pay a penalty of a factor of $\log(N)$ increase in the number of wavelengths used.

4.3 Waveband Switching for Multi-Source Traffic

We now consider the more general case of multi-source traffic. In this scenario, N nodes are connected to a central hub. Each node is assumed to have a hardware limitation of P transmitters and receivers, and can therefore send and receive up to P calls. The hub must switch the calls, at a band level, from the appropriate source to destination nodes. In our discussion, we will assume that self-traffic is allowed; the case without self-traffic is similar and leads to comparable results. We will show that many of the concepts in this scenario parallel those in the single-source case. (The primary differences are that the traffic set now consists of a traffic matrix rather than a vector, and the switching configuration for each waveband will now consist of a permutation matrix rather than a unit vector.) We will again begin by considering two special cases, the minimum-wavelength and minimum-waveband problems, followed by investigating algorithms that provide a tradeoff between these two cases.

4.3.1 The Minimum-Wavelength Problem

Recall that for the minimum-wavelength problem, we constrain ourselves to the domain of banding algorithms which use only the minimum possible number of wavelengths. Since each node may send up to P calls, it is clear that at least P wavelengths are necessary. In [32] it is shown that this is also sufficient. The challenge is therefore to first partition the P wavelengths into wavebands, and second, to develop an algorithm that will provide a valid wavelength assignment for any admissible traffic set using these wavebands.

We first address the partitioning of the wavebands. We consider the cases of interest to be maximal traffic sets, since we can add fictitious calls to any non-maximal set to construct a maximal one. We define a wavelength to be *fully utilized* if it is used to carry a call on every link. Mathematically, this is equivalent to stating that the matrix of calls supported by that wavelength forms a permutation matrix. We say that a waveband is fully utilized if every wavelength in that waveband is fully utilized.

Every minimum-wavelength algorithm must be able to fully utilize every waveband under any admissible maximal traffic set. We have already seen in Section 4.2.1 that a greedy approach is optimal for this type of problem. Recall that the greedy algorithm worked recursively by partitioning, at each step, the largest waveband size possible subject to the constraint that it could be fully utilized by any admissible traffic set. Intuitively, this was because the number of wavebands required turned out to be non-increasing in the amount of traffic; therefore at each step the optimal solution was to minimize the residual amount of traffic. The only remaining problem is that largest fully utilizable waveband size, $b_{max}(N, P)$, must be derived for the multi-source case.

We can view $b_{max}(N, P)$ as the largest number of identical permutation matrices that we are guaranteed to be able to find within *any* $N \times N$ matrix with row and column sums equal to P. This is equivalent to stating that sufficiently many calls must exist within any admissible traffic set with parameters N and P to fully utilize the waveband.

Example 8. *Consider the case where $P = 9$ and $N = 3$. One admissible maximal traffic set satisfying these constraints is given by*

$$C_1 = \begin{bmatrix} 1 & 5 & 3 \\ 3 & 2 & 4 \\ 5 & 2 & 2 \end{bmatrix}$$

The largest waveband that could be fully utilized by this particular traffic set is 4 wavelengths, as shown below:

$$C_1 = 4 \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 3 \\ 3 & 2 & 0 \\ 4 & 2 & 2 \end{bmatrix}$$

The first term consists of 4 identical permutation matrices representing calls that can be used to fully utilize a waveband of size 4 or less; the second term forms the remaining calls. Note that this shows only that b_{max} must be at most *4; it may be possible that some other traffic set may require an even smaller waveband for full utilization. For example, consider an alternate admissible set, formed by:*

$$C_2 = \begin{bmatrix} 2 & 4 & 3 \\ 4 & 2 & 3 \\ 3 & 3 & 3 \end{bmatrix}$$

In this example, since no entry in the third row contains more than 3 calls, at best 3 identical permutations can be found, with one possible assignment shown below:

$$C_2 = 3 \cdot \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 3 \\ 3 & 3 & 0 \end{bmatrix}$$

Therefore at best the largest waveband b_{max} can consist of 3 wavelengths; any larger waveband could not be fully utilized by C_2. Note that this is still not enough to show that b_{max} might not need to be lowered further; it may be possible that some other traffic set may require an even smaller waveband for full utilization. The challenge will be to obtain in closed form an equation for b_{max} without performing this sort of exhaustive examination of all possible maximal admissible traffic sets.

We will use results from graph theory to help derive a solution to this problem. We can represent any admissible traffic set as a bipartite graph. The bipartite graph consists of two sets of N nodes each. Denote these sets by $V_1 = \{s_1, \ldots, s_N\}$ and $V_2 = \{d_1, \ldots, d_N\}$. Nodes in V_1 represent sources, and nodes in V_2 represent destinations. We also create a set of edges $\mathcal{E}$ that is dependent on the traffic set under consideration: if there is at least one call from a source node i to a destination node j, create an edge connecting node s_i to node d_j. The edge is given a weight equal to the number of calls between that source-destination pair.

Define a *maximal matching* to be a subset of edges from $\mathcal{E}$ such that exactly one edge is incident on each node in V_1 and V_2. In our original matrix notation, this corresponds to a matrix T where entry $[T]_{i,j}$ is nonzero if and only if an edge between s_i and d_j is contained in the matching. Note that T is a permutation matrix, since there is only one nonzero entry per row (since only one edge is adjacent to each s_i) and column (due to one edge per d_j). The number of such permutation matrices that can be found in the particular traffic set is equal to the smallest edge weight in the maximal matching.

Example 9. *We consider again the set C_1 given in Example 8. Figure 4-3(a) gives the bipartite graph corresponding to C_1, and Figure 4-3(b) locates a bipartite matching*

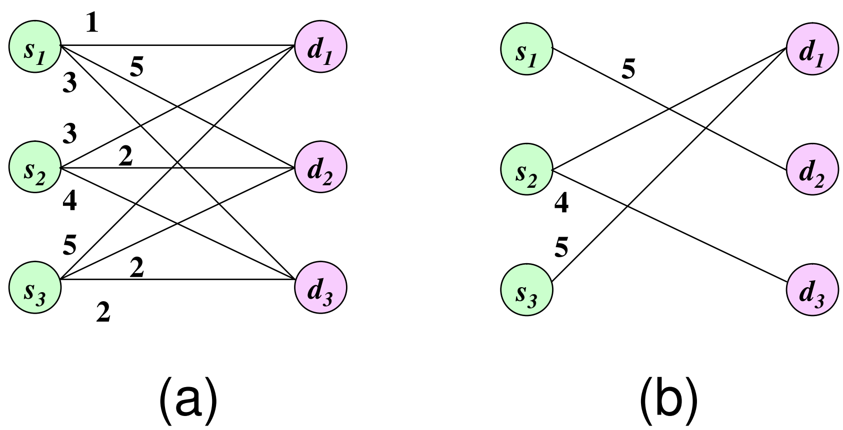


Figure 4-3: (a) The bipartite graph corresponding to C_1 in Example 9. (b) One possible bipartite matching from the graph.

within this graph. The bipartite matching corresponds to the permutation matrix

$$T_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Note that as the smallest edge weight on the matching is 4, we are able to locate 4 copies of T in C_1, as expected.

Therefore, in this graph context, we wish to choose b_{max} such that in any bipartite graph corresponding to an admissible traffic set, a maximal matching can be found where the smallest-weight edge is at least b_{max}. Conveniently, a theorem exists which provides necessary and sufficient conditions for the existence of maximal matchings on bipartite graphs.

Hall's Theorem [37]*:* In a bipartite graph $(V_1, V_2, \mathcal{E})$, define the neighborhood of a subset $v \subset V_1$ to be those nodes in V_2 which are connected via some edge in $\mathcal{E}$ to some node in v. Then there exists a maximal matching if and only if, for every subset $v \subset V_1$, its neighborhood $N(v)$ has size $|N(v)| \geq |v|$.

Hall's Theorem provides the basis for determining the existence of maximal matchings. The following test is applied: a subset v of the source nodes V_1 is chosen. If

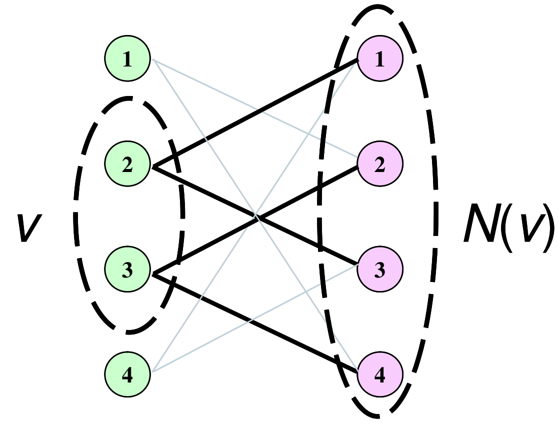


Figure 4-4: To apply the test given by Hall's Theorem, a subset v of the source nodes V_1 is first chosen. In this case, v consists of 2 nodes, and the neighborhood $N(v)$ contains 4 nodes. Therefore the test is passed for this choice of v. This test must be repeated for all possible choices of v.

the neighborhood of the subset $N(v)$ is of size greater than or equal to the size of the subset itself, the test is passed. This is shown in Figure 4-4. The test is then repeated for all possible subsets v of V_1. If the test is passed for all subsets, then a maximal matching exists. If at least one test is failed, then no maximal matching exists.

We can determine if a waveband of a given size b can be fully utilized by a given traffic set as follows. Determine the bipartite graph corresponding to the traffic set, and delete any edges with weight less than b. This removes from consideration any maximal matchings with edge weight less than b, since such matchings cannot fully utilize the waveband. The tests given by Hall's Theorem can then be applied to this graph to determine if a maximal matching exists that is sufficiently large to fully utilize the waveband. If the test fails, then a waveband of size b is too large to be sufficiently utilized. This test should be applied to all maximal traffic sets to guarantee that *any* maximal set can fully utilize the waveband.

In principle, the preceding approach could be used to determine $b_{max}(N, P)$ numerically by brute force. However, we will see that a closed-form solution can be obtained. The method for obtaining the closed-form expression for $b_{max}(N, P)$ relies on attempting to construct a bipartite graph which causes the test given by Hall's Theorem to be failed. (In a slight abuse of notation, we call such a bipartite match-

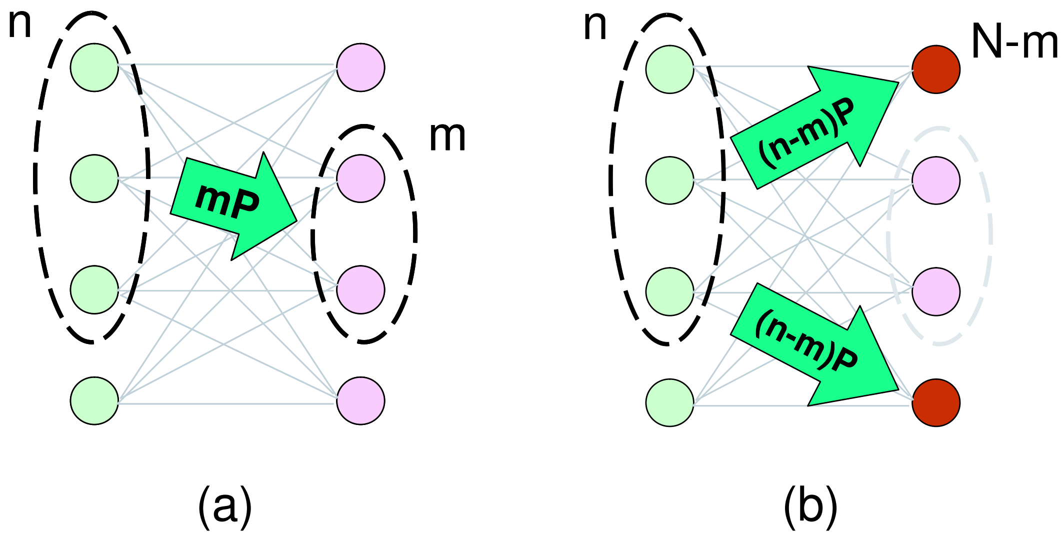


Figure 4-5: At most mP calls can be sent to nodes in $N(v)$, and hence $(n-m)P$ calls must go to non-neighborhood nodes.

ing a "counterexample".) $b_{max}(N, P)$ is then the largest waveband size for which no counterexample exists.

In order for the test to be failed, a maximal traffic set must be found for which we can choose a v such that the size of the neighborhood $N(v)$ is smaller than the size of v. We therefore wish to construct a counterexample where, if $|v| = n$, then $|N(v)| = m$, where $m < n$.

Under the P-port model, the nodes in v can send at most mP calls to nodes in $N(v)$. The remaining *residual traffic* is therefore at least $(n - m)P$. These calls are sent to nodes outside the neighborhood, and hence must belong to edges adjacent to a non-neighborhood node. Call these edges *non-neighborhood edges*. Non-neighborhood edges are those edges with weight less than $b_{max}(N, P)$ which are not eligible to belong to a maximal matching, since they do not contain enough calls to fully utilize the matching. This is illustrated in Figure 4-5. There are at most $n \cdot (N - m)$ non-neighborhood edges.

We can therefore construct a counterexample if and only if the residual traffic can be divided among the non-neighborhood edges such that no non-neighborhood edge has weight greater than or equal to $b_{max}(N, P)$. Since there are $(n - m)P$ residual calls and $n(N-m)$ non-neighborhood edges, there is at least one non-neighborhood edge with weight at least

$$\left\lceil \frac{(n-m)P}{n(N-m)} \right\rceil$$

If $b_{max}(N,P)$ is chosen to be at most this number, then no counterexample exists for the given values of n and m. We can choose $b_{max}(N,P)$ to guarantee that *no* counterexample exists by minimizing over n and m and choosing $b_{max}(N,P)$ smaller than this minimum:

$$\begin{aligned} b_{max} &\leq \min_{n,m} \left\lceil \frac{(n-m)P}{n(N-m)} \right\rceil \\ &= \min_{n,m} \left\lceil \frac{\left(1-\frac{m}{n}\right)P}{N-m} \right\rceil \end{aligned} \tag{4.6}$$

We fix for the moment m and consider the minimization over n. Since $n > m$, the minimization is subject to the constraint $0 < m < n < N$. Equation 4.6 is minimized by choosing n as small as possible. Since n and m are both integer quantities, we should choose $n = m + 1$; conversely, $m = n - 1$.

Making this substitution, the minimization becomes:

$$b_{max} \leq \min_{n} \left\lceil \frac{P}{n[N-(n-1)]} \right\rceil \tag{4.7}$$

Since the ceiling function is monotonic,

$$\min_{n} \left\lceil \frac{P}{n[N-(n-1)]} \right\rceil = \left\lceil \min_{n} \frac{P}{n[N-(n-1)]} \right\rceil$$

Ignoring integrality constraints, the right-hand size is easily shown to be minimized at $n^* = \frac{N+1}{2}$. If N is odd, then $\frac{N+1}{2}$ is an integer and hence is a valid choice for n^*. We subsequently obtain a value for b_{max} of

$$\begin{aligned} b_{max} &= \left\lceil \frac{P}{\left(\frac{N+1}{2}\right)\left(\frac{N+1}{2}\right)} \right\rceil \\ &= \left\lceil \frac{4P}{(N+1)^2} \right\rceil \quad , \quad \text{if } N \text{ odd} \end{aligned}$$

If N is even, then since $\frac{P/n}{N-(n-1)}$ is convex, the minimizing value of n must be one of the integers adjacent to $\frac{N+1}{2}$, namely either $\lfloor \frac{N+1}{2} \rfloor = \frac{N}{2}$ or $\lceil \frac{N+1}{2} \rceil = \frac{N}{2} + 1$. It is easy to verify that either case results in the same value of

$$\begin{aligned} b_{max} &= \left\lceil \frac{P}{\left(\frac{N}{2}\right)\left(\frac{N}{2}+1\right)} \right\rceil \\ &= \left\lceil \frac{4P}{N(N+2)} \right\rceil \quad , \quad \text{if } N \text{ even} \end{aligned}$$

In summary,

$$b_{max} = \begin{cases} \left\lceil \frac{4P}{N(N+2)} \right\rceil & , N \text{ even} \\ \\ \left\lceil \frac{4P}{(N+1)^2} \right\rceil & , N \text{ odd} \end{cases} \tag{4.8}$$

Now that $b_{max}(N, P)$ is known for general P-port traffic, (4.8) can be used in conjunction with the greedy algorithm to optimally partition the wavebands. Again, note that the waveband sizing is independent of the particular traffic set and depends only on the network parameters N and P. Therefore wavebands of constant size can be used for any admissible traffic set. The assignment of calls to the wavebands can be obtained by a bipartite matching algorithm; by using the expression $b_{max}(N, P)$ to size each waveband, we have guaranteed that maximal matchings can be found which fully utilize each waveband for any admissible traffic set.

Example 10. *We continue with the 3-node network from Example 8 where* $P = 9$. *Using the greedy algorithm, we would determine that the largest waveband should be* $\left\lceil \frac{4P}{(N+1)^2} \right\rceil = \left\lceil \frac{(4)(9)}{(4)^2} \right\rceil = 3$.

After this step, $9 - 3 = 6$ *wavelengths remain to be partitioned. We repeat this process until all wavelengths have been assigned to bands. The final waveband partition is* $\{3, 2, 1, 1, 1, 1\}$. *By choice of* b_{max}, *this partitioning can be fully utilized by any admissible traffic set. For example, consider the traffic set* C_1 *from Example 8. One possible decomposition is*

$$\begin{bmatrix} 1 & 5 & 3 \\ 3 & 2 & 4 \\ 5 & 2 & 2 \end{bmatrix} = 3 \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} + 2 \cdot \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$
$$+1 \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + 1 \cdot \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$+1 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 1 \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Note that the equality in this decomposition indicates that all wavelengths are fully utilized. Furthermore, by the optimality of the greedy algorithm, we are guaranteed that this is the minimum possible number of wavebands subject to the minimum-wavelength constraint.

Notice that there are $\frac{(N+1)^2}{4} - 1 = 3$ *wavebands consisting of a single wavelength. This results from the fact that once* P *becomes less than the denominator, the greedy algorithm will choose all remaining wavebands to be of size 1. This indicates that banding is only useful when* $P > N^2/4$ *if no wavelength inefficiency is allowed. Essentially, this fact is based on the intuition that if the number of nodes is large relative to the number of calls, then we cannot guarantee that sufficiently many calls will belong to the same source-destination pairs for banding to take place.*

In principle, with the greedy algorithm, we can obtain the exact minimum number of wavebands required under the minimum-wavelength constraint simply by iterating through the algorithm and counting the number of wavebands produced. We can also obtain in closed form an upper bound on the number of wavebands by relaxing the integer constraints and using arguments analogous to the single-source case. Let P_k be the value of P after running the k^{th} iteration of the greedy algorithm. The series progresses as follows:

$$\begin{aligned} P_1 &= P - \frac{4}{(N+1)^2} \cdot P = \left[1 - \frac{4}{(N+1)^2}\right] \cdot P \\ P_2 &= \left[1 - \frac{4}{(N+1)^2}\right] \cdot P_1 = \left[1 - \frac{4}{(N+1)^2}\right]^2 \cdot P \\ &\vdots \\ P_k &= \left[1 - \frac{4}{(N+1)^2}\right]^k \cdot P \end{aligned} \tag{4.9}$$

If $P \leq \frac{(N+1)^2}{4}$, then the number of bands B is simply equal to $\frac{(N+1)^2}{4}$ since each band is composed of only a single wavelength. Therefore consider $P > \frac{(N+1)^2}{4}$ and determine the number of bands k required to reduce the number of unassigned wavelengths to $\frac{(N+1)^2}{4}$. Then the total number of wavebands would be $k + \frac{(N+1)^2}{4}$.

$$\begin{aligned} P_k &= \frac{(N+1)^2}{4} \\ \left[1 - \frac{4}{(N+1)^2}\right]^k \cdot P &= \frac{(N+1)^2}{4} \\ k &= \frac{\log\left[\frac{(N+1)^2}{4P}\right]}{\log\left[1 - \frac{4}{(N+1)^2}\right]} \end{aligned}$$

Again, since removing the ceiling constraint underestimates the size of each waveband, this gives an upper bound on the number of wavebands B, namely:

$$B \leq \begin{cases} \frac{(N+1)^2}{4} + \frac{\log\left[\frac{(N+1)^2}{4P}\right]}{\log\left[1-\frac{4}{(N+1)^2}\right]} & , \; P > \frac{(N+1)^2}{4} \\ \\ P & , \; P \leq \frac{(N+1)^2}{4} \end{cases} \tag{4.10}$$

It can be show from (4.10) that the number of wavebands required grows as $O(N^2 \log(P/N^2))$. Since the number of wavebands required by the greedy algorithm is the minimum possible for any minimum-wavelength algorithm, this allows us to quantify the maximum switching reduction possible without wavelength inefficiency.

Example 11. *Recall the case of Example 10, where $N = 3$ and $P = 9$. From (4.10), we obtain an upper bound on the number of wavebands of 6.82. Since the number of wavebands must be an integer, we can safely round the bound down to 6, which is exactly equal to the true value of 6 wavebands. We can also use (4.10) to approximate the number of wavebands by 5.4, which is still close to the true value of 6 wavebands.*

4.3.2 The Minimum-Waveband Problem

Recall that a minimum-waveband algorithm is defined to be a banding algorithm that uses the minimum possible number of wavebands. Since all wavelengths in the same waveband must go to the same destination, and there are N possible different destinations, a minimum of N wavebands are required. One way to achieve this is to statically provision P wavelengths between each source-destination pair, using a total of PN wavelengths. The minimum-waveband problem is therefore to find a better, dynamic algorithm that uses fewer wavelengths. Since the single-source traffic model is a special case of the multi-source model, we can also use (4.5) to provide a lower bound on the number of wavelengths required:

$$W \geq \sum_{i=1}^{N} \left\lfloor \frac{P - N + i}{i} \right\rfloor$$

However, in this case it is possible to show that the bound is not tight. We therefore do not know the achievable minimum number of wavelengths, only that it cannot be less than that specified by (4.5).

We next propose a wavelength-efficient minimum-banding algorithm which requires $O(P\sqrt{N})$ total wavelengths, which improves on the $O(PN)$ worst case. The algorithm operates by decomposing the traffic set into N sub-matrices, and attempts to group entries with heavy weights and light weights into separate sub-matrices. This will allow some wavebands to use less than the worst case of P. The algorithm relies on the following lemma:

Lemma 13. *Consider a P-port traffic set on an N-node star. For any value of k such that $1 \leq k \leq N$, there exists a decomposition satisfying (4.1) where*

$$\begin{aligned} b_1 &= \ldots = b_k = P \\ b_{k+1} &= \ldots = b_N = \left\lceil \frac{P}{k+1} \right\rceil \end{aligned}$$

Proof. The proof of Lemma 13 will be by construction. We will first decompose the traffic matrix C into two sub-matrices: the "heavy" matrix C_H, containing all entries with weight greater than $\lceil \frac{P}{k+1} \rceil$, and the "light" matrix C_L, containing *no* entries greater than $\lceil \frac{P}{k+1} \rceil$.

We first assign any entry in C greater than $\lceil \frac{P}{k+1} \rceil$ to C_H. Note that at this point each row and column in C_H contains at most k entries. (If any row or column exceeds k entries, then that row or column in C must have had a sum greater than P, meaning C is not an admissible traffic set.) We next continue assigning entries in C to C_H until each row and column of C_H has exactly k entries.

Suppose there exists a row in C_H that contains fewer than k entries. Then there must also be a column that has fewer than k entries. (This follows from the fact that C_H is square; if all columns have k entries, and each row has no more than k entries, then all rows must also have k entries.) Locate the entry corresponding to that row and column in C, and assign it to C_H. Repeat until each row has k entries.

By the same reasoning as before, all columns must now have k entries also. It is well known that any such matrix can be decomposed into at most k matrices with only one non-zero entry per row and column. Therefore, by performing this further decomposition and noting that all entries in C_H are at most P, we have shown that C_H can be supported by at most k wavebands of size P.

Assign all remaining entries in C to C_L; this gives C_L therefore has exactly $N-k$ entries per row and column. This can similarly be decomposed into $N-k$ matrices with only one entry per row and column; since each entry is at most $\lceil \frac{P}{k+1} \rceil$, we can support C_L using at most $N-k$ wavebands of size $\lceil \frac{P}{k+1} \rceil$. □

Corollary: Any P-port N-node traffic set can be routed using k bands of size P and $N-k$ bands of size $\lceil \frac{P}{k+1} \rceil$.

The proof of Lemma 13 forms the basis of the SQRT(N) algorithm. Since it holds for any value of k, it is logical to use the value of k which results in the fewest total number of wavelengths used. To determine this, we write down the expression for the number of wavelengths required:

$$W_k = kP + (N-k)\left(\frac{P}{k+1}\right) \tag{4.11}$$

It can be shown that this expression is minimized at $k = \sqrt{N+1} - 1$. If we relax the integer constraint on k and substitute this back into the equation, we obtain:

$$W = 2P\left(\sqrt{N+1} - 1\right)$$

leading to the observation that the SQRT(N) algorithm requires $O(P\sqrt{N})$ wavelengths. The results of this section show that the maximum amount of switching reduction can be achieved by, at worst, a factor of $\sqrt{N}$ increase in the number of wavelengths.

Example 12. *We examine the case of Example 10 under the minimum-waveband* *restriction. Using the SQRT(N) algorithm, we see that for* $N = 3$ *and* $P = 9$*, we should choose* $k = \sqrt{N+1} - 1 = 1$*. We therefore require only 1 waveband of size 9 and* $N - 1 = 2$ *bands of size* $\lceil P/(k+1) \rceil = 5$*, producing a final waveband sizing of* $\{9, 5, 5\}$*, for a total of 19 wavelengths and 3 bands. We can compare this with the optimal minimum-wavelength solution of* $\{3, 2, 1, 1, 1, 1\}$*, which uses the minimum of 9 wavelengths but requires 6 wavebands.*

Shown below is one possible call assignment using these waveband sizes for the specific traffic set C_1*.*

$$\begin{bmatrix} 1 & 5 & 3 \\ 3 & 2 & 4 \\ 5 & 2 & 2 \end{bmatrix} \leq 9 \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} + 5 \cdot \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$
$$+5 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 9 & 5 \\ 5 & 5 & 9 \\ 9 & 5 & 5 \end{bmatrix}$$

Note that wavelengths were overprovisioned in this case for some source-destination pairs. However, the additional wavelengths are required to guarantee that all admissible traffic sets can be accommodated by the 3 wavebands.

4.3.3 Hybridization: Wavelength-Waveband Tradeoffs

The greedy algorithm previously discussed provides us with one point on the optimal frontier. The SQRT(N) algorithm provides us with a second point which, while not necessarily optimal, nonetheless provides good performance in the minimum-waveband case. As in the single-source traffic case, we can now investigate the use of a dual-algorithm approach to obtaining a tradeoff between these two performance points.

To review, in the dual algorithm approach, the initial g wavebands are allocated using the greedy algorithm. The motivation for this is that the greedy algorithm performs well when the number of calls is large relative to the number of nodes. The remaining calls are allocated using the SQRT(N) algorithm, which always uses the minimum number of wavebands N. By varying the number of wavebands g given over to the greedy algorithm, more emphasis can be given to reducing either wavelengths or wavebands. At $g = 0$, the dual-algorithm approach is reduced to exactly the SQRT(N) algorithm; conversely for g sufficiently large, only the greedy algorithm will be used.

Example 13. *Consider a 10-node star with* $P = 1000$. *Suppose the greedy algorithm is used to assign the first* $g = 25$ *wavebands, and the remaining calls use the SQRT(N) algorithm. Using the equation for* $b_{max}(N, P)$ *given by (4.8), the greedy algorithm chooses waveband sizes of* {*34, 33, 32, 31, 29, 29, 28, 27, 26, 25, 24, 23, 22, 22, 21, 20, 20, 19, 18, 18, 17, 17, 16, 15, 15*}. *Summing the band sizes shows that a total of 581 wavelengths are used.*

After this assignment, 419 calls per node remain. Using the SQRT(N) algorithm, 10 additional wavebands are required to accommodate them. Since $\sqrt{N+1} - 1 = 2.317$, *the minimizing value of* k *must be either 2 or 3; substituting both values into (4.11) shows that* $k = 2$ *is optimal. Therefore we choose 2 wavebands of size 419, and 8 wavebands of size* $\lceil 419/(2+1) \rceil = 140$, *for a total of 1958 wavelengths.*

Therefore in this example a total of 35 wavebands, containing $1958 + 581 = 2539$ *wavelengths, are required.*

We can obtain an approximate expression for the tradeoff curve from the dual-algorithm approach as follows. The greedy algorithm is initially employed for g iterations. From (4.9), we know that after g iterations, the remaining traffic is given by

$$P' = \left[1 - \frac{4}{(N+1)^2}\right]^g \cdot P$$

The greedy algorithm is therefore responsible for using g wavebands, of total size $P - P'$. The remaining traffic is handled using the SQRT(N) algorithm, which uses N wavebands and $= 2P'\left(\sqrt{N+1} - 1\right)$ wavelengths. Therefore the total number of wavelengths and wavebands used can be expressed by

$$
\begin{aligned}
W_{dual} &= W_{greedy} + W_{min-band} \\
&= \left(1 - \left[1 - \frac{4}{(N+1)^2}\right]^g\right) \cdot P \\
&\quad + \left[1 - \frac{4}{(N+1)^2}\right]^g \cdot 2P\left(\sqrt{N+1} - 1\right) \\
&= \left[1 + \left(2\sqrt{N+1} - 3\right)\left(1 - \frac{4}{(N+1)^2}\right)^g\right] \cdot P \\
&\approx \left[1 + 2\sqrt{N}\left(1 - \frac{4}{(N+1)^2}\right)^g\right] \cdot P \\
B_{dual} &= B_{greedy} + B_{min-band} \\
&= g + N
\end{aligned}
$$

Combining, we obtain

$$
W_{dual} = \left[1 + 2\sqrt{N}\left(1 - \frac{4}{(N+1)^2}\right)^{B_{dual}-N}\right] \cdot P
$$

The dual-algorithm approach actually produces a family of algorithms. Choosing the number of wavebands B specifies the number of wavelengths W, and vice versa. Figure 4-6 shows the performance of this approach for a star with $N = 10$, $P = 1000$. The two asymptotes represent the SQRT(N) and greedy algorithms, which operate in the minimum-waveband and minimum-wavelength regimes, respectively. The dual-algorithm approach essentially interpolates between the minimum-waveband and minimum-wavelength cases to produce algorithms with intermediate waveband and wavelength requirements.

4.4 The Uniform Waveband Approach

Thus far in discussing multi-source traffic, we have allowed wavebands to be non-uniformly sized. Here we consider fixing all wavebands to a constant size b and

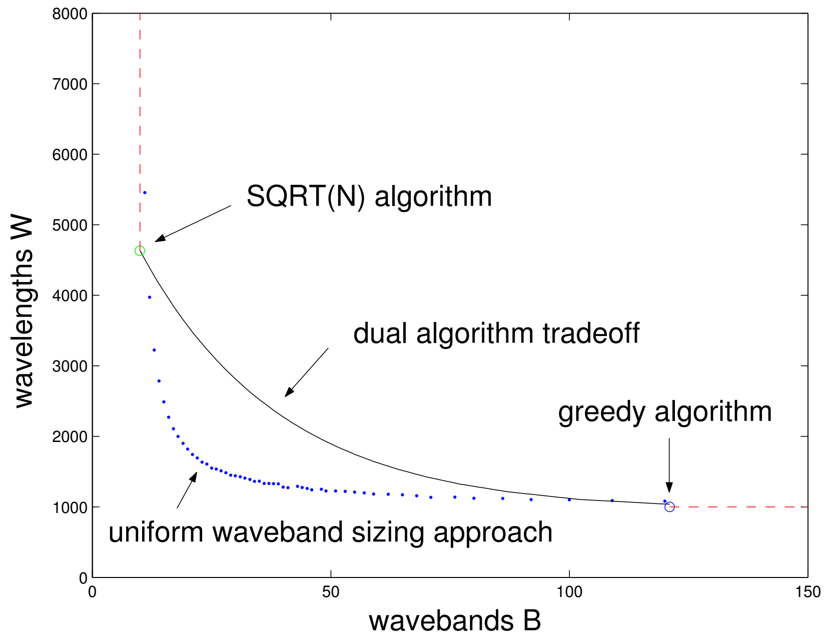


Figure 4-6: Performance of the uniform waveband algorithm for a star with $N = 10$, $P = 1000$. The red and blue asymptotes represent the SQRT(N) and greedy algorithms, respectively.

derive the number of wavelengths and wavebands required. By varying b, a family of banding algorithms with varying numbers of wavebands and wavelengths can be obtained. Somewhat surprisingly, we will see that in the multi-source case, by using uniform waveband sizes, most of the maximum banding gain can be obtained at a very small cost in additional wavelengths.

We begin by first deriving the minimum number of wavebands required for a fixed waveband size b.

Theorem 10. *Given a fixed band size b, the necessary and sufficient minimum number of wavebands required to support P-port traffic in an N-node star is*

$$B_{uniform} = N + \left\lfloor \frac{P-N}{b} \right\rfloor \tag{4.12}$$

Corollary: The corresponding necessary and sufficient minimum number of wavelengths required is

$$W_{uniform} = bN + b\left\lfloor \frac{P-N}{b} \right\rfloor \tag{4.13}$$

We first prove necessity of (4.12) by providing an example which requires at least this number of wavelengths. Sufficiency will be shown by construction. (4.13) then follows directly from the fact that each band is of size b.

Consider the traffic set where node 1 sends a single call to nodes 1 to $N-1$, and $P-(N-1)$ calls to node N. In this case, $N-1$ bands are required to support traffic to the first $N-1$ nodes, while $\left\lceil \frac{P-(N-1)}{b} \right\rceil$ bands are required to support traffic to node N. This gives a lower bound on the number of wavebands of

$$\begin{aligned} B_{uniform} &\geq (N-1) + \left\lceil \frac{P-(N-1)}{b} \right\rceil \\ &= N + \left\lfloor \frac{P-N}{b} \right\rfloor \end{aligned}$$

where the last step follows from the observation that $\left\lceil \frac{P-(N-1)}{b} \right\rceil = \left\lfloor \frac{P-N}{b} \right\rfloor + 1$.

We will show using a bipartite matching approach that these quantities are sufficient as well. We will first construct a bipartite multigraph. A multigraph differs from a graph in that it allows multiple edges between the same two nodes. We consider two sets of nodes $V_1 = \{s_1, \ldots, s_N\}$ and $V_2 = \{d_1, \ldots, d_N\}$. For a given admissible traffic set, define the number of calls from node i to node j to be $c_{i,j}$. Then create $\lceil \frac{c_{i,j}}{b} \rceil$ edges connecting node s_i to d_j. The complete set of edges $\mathcal{E}$ represents the traffic to be carried, now in units of wavebands (each of size b) instead of wavelengths.

The number of edges adjacent to each source node s_i can be obtained by summing $\lceil \frac{c_{i,j}}{b} \rceil$ over j. Let the number of destinations receiving non-zero traffic from node i be N_i. Without loss of generality, assume these are nodes 1 through N_i. We decompose the traffic to these destinations by

$$c_{i,j} = \omega_{i,j} + r_{i,j} \tag{4.14}$$

where $r_{i,j}$ is chosen such that $\omega_{i,j}$ is a nonnegative integer multiple of b, and $1 \leq r_{i,j} \leq b$. We can then express the summation of interest as

$$\begin{aligned}
\sum_{i=1}^{N} \left\lceil \frac{c_{i,j}}{b} \right\rceil &= \sum_{i=1}^{N} \left\lceil \frac{\omega_{i,j} + r_{i,j}}{b} \right\rceil \\
&= \sum_{i=1}^{N_i} \left\lceil \frac{r_{i,j}}{b} \right\rceil + \sum_{i=1}^{N_i} \frac{\omega_{i,j}}{b} \\
&= N_i + \frac{1}{b} \sum_{i=1}^{N_i} \omega_{i,j} \qquad (4.15)
\end{aligned}$$

where the second step relies on $\frac{\omega_i}{b}$ being integer, and the third on the fact that $r_i \leq b$.

By summing (4.14) over i and noting that $\sum_{i=1}^{N_i} c_{i,j} = P$, we can obtain the following useful relation:

$$\begin{aligned}
P &= \sum_{i=1}^{N_i} \omega_{i,j} + \sum_{i=1}^{N_i} r_{i,j} \\
\Rightarrow \sum_{i=1}^{N_i} \omega_{i,j} &= P - \sum_{i=1}^{N_i} r_{i,j} \\
&\leq P - N_i
\end{aligned}$$

where the last line results from observing that $r_{i,j} \geq 1$. Using this result, (4.15) becomes

$$
\begin{aligned}
\sum_{i=1}^{N} \left\lceil \frac{c_{i,j}}{b} \right\rceil &\leq N_i + \frac{P - N_i}{b} \\
&= \left(1 - \frac{1}{b}\right) N_i + \frac{P}{b} \\
&\leq \left(1 - \frac{1}{b}\right) N + \frac{P}{b} \\
&= N + \frac{P - N}{b}
\end{aligned}
$$

Since both the summation on the left and N on the right are integers, by taking the floor of both sides we can conclude

$$
\sum_{i=1}^{N} \left\lceil \frac{c_{i,j}}{b} \right\rceil \leq N + \left\lfloor \frac{P - N}{b} \right\rfloor
$$

Similar arguments can be used to show that the number of edges adjacent to each destination node d_j is given by

$$
\sum_{j=1}^{N} \left\lceil \frac{c_{i,j}}{b} \right\rceil \leq N + \left\lfloor \frac{P - N}{b} \right\rfloor
$$

Therefore each node has at most $N + \left\lfloor \frac{P-N}{b} \right\rfloor$ edges adjacent to it. The following lemma will now prove useful.

Lemma 14. *In a bipartite multigraph where each node is adjacent to at most k edges, a partitioning exists that divides the edges into at most k matchings.*

Proof. See [37]. □

By Lemma 14, at most an equal number of matchings are required. Since calls in each matching can share the same waveband, at most $N + \left\lfloor \frac{P-N}{b} \right\rfloor$ wavebands are required. The number of wavelengths follows directly from the fact that each waveband is of size b.

Example 14. *Consider a star with* $N = 10$ *and* $P = 1000$*, and consider uniform band sizes of* $b = 40$*. By Theorem 10,* $N + \lfloor \frac{P-N}{b} \rfloor = 10 + \lfloor \frac{1000-10}{50} \rfloor = 29$ *wavebands are required. Since each waveband consists of* 50 *wavelengths, a total of* $50 \cdot 29 = 1711$ *wavelengths are used as well.*

We now have a method for obtaining necessary and sufficient conditions on the number of wavebands and wavelengths are necessary and sufficient. By ignoring the integrality constraints, we can solve for the number of wavelengths as a function of the number of wavebands and obtain an approximate characterization of the waveband-wavelength tradeoff using uniformly-sized wavebands:

$$W_{uniform} = P + \left(\frac{P-N}{B_{uniform} - N} - 1 \right) N \tag{4.16}$$

Figure 4-6 illustrates the performance of the uniform waveband algorithm for a star with $N = 10$, $P = 1000$. The two asymptotes represent the SQRT(N) and greedy algorithms, which operate in the minimum-waveband and minimum-wavelength regimes, respectively. Note that although the uniform waveband algorithm performs poorly in the minimum-waveband regime (where it requires many more wavelengths than the SQRT(N) algorithm), the performance improves dramatically once a few additional wavebands are introduced. By around 40 wavebands, it requires only slightly more wavelengths than the greedy algorithm, which uses 121 wavebands. We observe that by allowing slightly more wavelengths than the minimum-wavelength case, the fixed-waveband algorithm can greatly reduce the number of wavebands required, approaching the minimum-waveband bound significantly.

From this graph, two observations can be made:

1. As the number of wavebands increases, the performance of the uniform-waveband algorithm appears to approach the optimal performance of the greedy algorithm. In particular, at the right endpoint, it appears to be almost wavelength-efficient.

2. Because of the slow increase in the number of wavelengths required as the number of wavebands decreases, it appears that the majority of the reduction

in the number of wavebands can be achieved at very little cost in wavelength inefficiency (as compared to the greedy algorithm).

The first observation can be verified by comparing the number of wavelengths used by the uniform-waveband algorithm to the greedy algorithm. For P large we can approximate B_{greedy} by $\frac{N^2}{4}\log\left(\frac{4P}{N^2}\right)$. For this number of bands, the number of wavelengths used by the uniform-waveband algorithm is approximately

$$\begin{aligned} W_{uniform} &\approx P + \frac{PN}{B} \\ &= \left[1 + \frac{1}{\frac{N^2}{4}\log\left(\frac{4P}{N^2}\right)}\right] \cdot P \\ &= (1+\alpha) \cdot P \end{aligned}$$

where α is a term that goes to zero as P increases. Recall that the greedy algorithm, which was wavelength-efficient, uses the minimum of P wavelengths. Therefore the performance of the uniform-waveband algorithm approaches the optimum asymptotically in the minimum-wavelength regime.

It is also possible to show analytically by slope analysis that $W_{uniform}$ approaches its final value very quickly; this gives rise to the second observation, which is extremely significant from a practical perspective. If we are interested in building an actual implementation, it indicates that a majority of the gain from using banding can be achieved with very little wavelength inefficiency. For example, in the graph of Figure 4-6, the processing granularity can be reduced from 1000 wavelengths (without banding) to 30 wavebands, a reduction of 97%, at a cost of only a 50% increase in the number of wavelengths.

4.5 Banding on General Topologies

Thus far all our results have been for the star topology. In this section we extend the preceding banding results to general topologies for which routing algorithms for

P-port traffic are known.

Recall that we have shown that banding can be considered as a matrix decomposition problem, where for a given admissible traffic set C, our goal is to decompose it into the sum of a fixed number B of weighted permutation matrices:

$$C \leq b_1 T_1 + b_2 T_2 + \ldots + b_B T_B$$

where the band sizes $\{b_i\}$ and the total number of wavebands B are constant for all traffic sets. The goal was to minimize, over all possible admissible traffic sets, the two cost parameters corresponding to the number of wavebands B and the number of wavelengths $\sum_{i=1}^{B} b_i$.

In the star, each permutation T_i could be accommodated using a single waveband consisting of b_i wavelengths. This approach can be extended to other topologies in a straightforward manner, with the main difference being that each permutation T_i may now require multiple wavebands of size b_i to support it. In general, if the RWA algorithm requires $\phi(N)$ wavelengths for permutation traffic on the topology, and a banding algorithm is considered which uses B wavebands and W wavelengths on a star, then the extension of that banding algorithm to the new topology requires:

$$\begin{aligned} B_{total} &= B \cdot \phi(N) \\ W_{total} &= W \cdot \phi(N) \end{aligned}$$

Routing algorithms for permutation traffic exist in the literature for rings with [6] and without [24] conversion, trees [32], and torus networks [33].

For example, in the case of a ring, [24] provides an optimal RWA algorithm for the bidirectional ring topology without conversion using the minimum number of wavelengths. Specifically, it shows that $\lceil N/3 \rceil$ wavelengths are necessary and sufficient to support any single-port traffic set. Since T_i is a permutation matrix, it can be supported using $\lceil N/3 \rceil$ wavelengths. Each set of calls $b_i T_i$ from the decomposition of (4.17), can therefore be supported using $\lceil N/3 \rceil$ wavebands, each consisting of b_i wavelengths.

Using this approach, the entire traffic set T can be supported using B sets of wavebands, where each such set i consists of $\lceil N/3 \rceil$ wavebands of b_i wavelengths. The total numbers of wavebands and wavelengths are

$$
\begin{aligned}
B_{total} &= B \left\lceil \frac{N}{3} \right\rceil \\
W_{total} &= \left\lceil \frac{N}{3} \right\rceil \sum_{i=1}^{B} b_i = W \left\lceil \frac{N}{3} \right\rceil
\end{aligned}
$$

4.6 Chapter Summary

In this chapter, we considered waveband switching as a method for reducing overall network costs. We observed that for networks with a large number of wavelengths relative to the number of nodes, switching at a coarser waveband granularity is both intuitive and efficient in terms of a wavelength efficiency versus node complexity tradeoff.

We first derived lower bounds on the number of wavelengths and wavebands required, bounding the achievable region in the wavelength-waveband performance space. We provided wavelength-efficient algorithms that use the minimum possible number of wavebands, and showed that the optimal approach is to use a greedy algorithm. We also provided minimum-waveband algorithms that allow for small wavelength inefficiencies in return for reducing the number of wavebands down to just the nodal degree. We used these results to help characterize the optimal achievable performance frontier.

We also provided both a dual algorithm approach and a uniform waveband approach that compare very favorably to the optimal performance frontier and achieve large reductions in switching requirements at very little cost in wavelength inefficiency.

Finally, we extended our results to general topologies where permutation traffic routing algorithms are known.

Chapter 5

Hybrid Static-Dynamic Networks

In the preceding chapters, we have assumed that all wavelengths accessible to a node can be dynamically switched – that is, the source and destination of the call being carried by any wavelength can be changed as necessary during the operation of a network. This allows for great flexibility in the wavelength assignment, at the cost of requiring complex switching and reconfiguration equipment for each wavelength.

We have described in our introductory remarks in Section 1.4 an alternate approach, which is to *statically* provision some wavelengths for the exclusive use of each user. These static wavelengths, because they do not require any switching or reconfiguration capability, are notably less expensive to provide than dynamic wavelengths. An additional, smaller, common pool of shared dynamically switched wavelengths are then provisioned to support any traffic that exceeds the static provisioning for each user. In this chapter, we will analyze the provisioning and operating characteristics of such hybrid networks.

5.1 System Model

In the shared link context, we can consider each incoming-outgoing pair of fibers to be a different *user* of the link. Each lightpath request (which we will henceforth term a *call*) can therefore be thought of as belonging to the user corresponding to the incoming-outgoing fiber pair that it uses. We can similarly associate each static

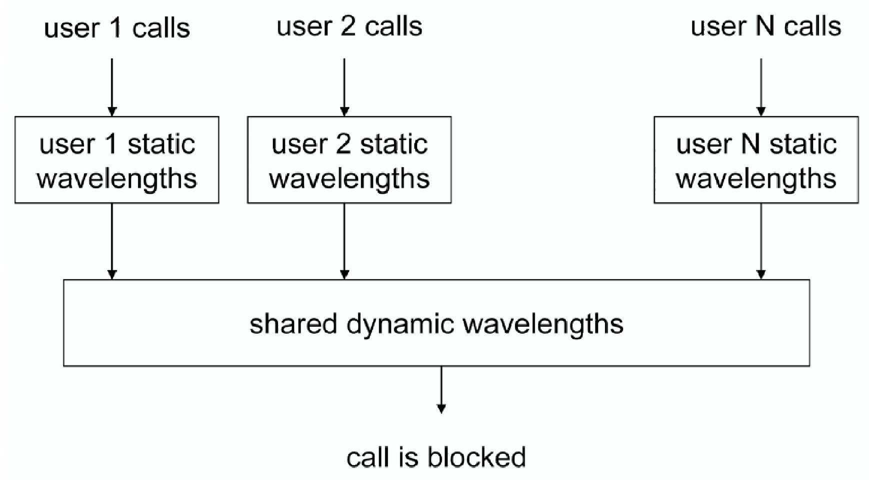


Figure 5-1: Decision process for wavelength assignment for a new call arrival. A new call first tries to use a static wavelength if it is available. If not, it tries to use a dynamic wavelength. If again none are available, then it is blocked.

wavelength with the corresponding user. Under these definitions, a call belonging to a given user cannot use a static wavelength belonging to a different user – it must either use a static wavelength belonging to its own user, or employ a dynamic wavelength.

Figure 5-1 gives a flowchart of the decision process for admitting a call. When a user requests a new call setup, the link checks to see if a static wavelength for that user is free. If there is a free static wavelength, it is used. If not, then the link checks to see if any of the shared dynamic wavelengths are free – if so, then a dynamic wavelength is used. If not, then no resources are available to support the call, and it is blocked.

There have been several approaches developed in the literature for blocking probability analysis of such systems under Poisson traffic models [15], including the Equivalent Random Traffic (ERT) model [38, 9, 23] and the Hayward approximation [13]. These approximations, while often able to produce good numerical approximations of blocking probability, are purely numerical in nature and do not provide good intuition for guiding the dimensioning of the wavelengths. Furthermore, they assume that the dynamic wavelengths must be individually switched, and do not consider waveband switching.

In this chapter, we adopt a snapshot traffic model that leads to closed-form asymptotic analysis and develop guidelines for efficient dimensioning of hybrid networks. We consider examining a "snapshot" of the traffic demand at some instant in time. The snapshot is composed of the vector $\mathbf{c} = [c_1, \ldots, c_N]$, where c_i is the number of calls that user i has at the instant of the snapshot, and N is the total number of users.

We model each variable c_i as a Gaussian random variable with mean μ_i and variance σ_i^2. This is reasonable since each "user" actually consists of a collection of source-destination pairs in the larger network that all use the link from the same source fiber to the same destination fiber. Although the traffic for each individual source-destination pair for the user may have some arbitrary distribution, as long as the distributions are well-behaved, the sum of each traffic stream will appear Gaussian by the Central Limit Theorem.

As a special case, consider the common model of Poisson arrivals and exponential holding times for calls. Then the number of calls that wish to enter the system at any instant in time is given by the stationary distribution of an $M/M/\infty$ queue – namely, Poisson with intensity equal to the load ρ in Erlangs. For a heavy load, this distribution is well approximated by a Gaussian random variable with mean ρ and variance ρ.

5.2 Wavelength-Granularity Switching

In this section, we consider a shared link, and assume that there are N users that are the source of calls on the link. Each user is statically provisioned W_s wavelengths for use **exclusively** by that user. In addition to this static provisioning, we will also provide a total of W_d dynamically switched wavelengths. These wavelengths can be shared by any of the N users.

As previously described, we will use a snapshot model of traffic. The traffic is given by a vector $\mathbf{c} = [c_1, \ldots, c_N]$, where each c_i is independent and identically distributed as $\mathsf{N}(\mu, \sigma^2)$. We assume that the mean μ is significantly large relative to σ that the probability of "negative traffic" (a physical impossibility) is low, and therefore does not present a significant modeling concern. We will primarily be concerned with a special blocking event that we call *overflow*. An overflow event occurs when there are insufficient resources to support all calls in the snapshot and at least one call is blocked. We will call the probability of this event the *overflow probability*.

From Figure 5-1, we see that an overflow event occurs if the total number of calls exceeds the ability of the static and dynamic wavelengths to support. This can be expressed mathematically as

$$\sum_{i=1}^{N} \max\{c_i - W_s\ ,\ 0\} > W_d \tag{5.1}$$

where $\max\{c_i - W_s\ ,\ 0\}$ is the amount of traffic from each user that exceeds the static provisioning; if the total amount of excess from each user exceeds the available pool of shared dynamic wavelengths, a blocking event occurs.

If we consider the N-dimensional vector space occupied by $\mathbf{c}$, the constraint given by (5.1) represents a collection of hyperplanes bounding the admissible traffic region:

$$\begin{aligned} c_i &\leq W_s + W_d \\ c_i + c_j &\leq 2W_s + W_d \quad , \quad i \neq j \\ c_i + c_j + c_k &\leq 3W_s + W_d \quad , \quad i \neq j \neq k \\ &\vdots \end{aligned}$$

Each constraint reflect the fact that the sum of the traffic from any subset of users clearly cannot exceed the sum of the static provisioning for those users plus the entire dynamic provisioning available. Note that there are a total of N sets of constraints, where the n^{th} set consists of $C(N,n) = \frac{N!}{(N-i)!n!}$ equations, each involving the sum of n elements of the traffic vector $\mathbf{c}$. If the traffic snapshot $\mathbf{c}$ falls within the region defined by the hyperplanes, all calls are admissible; otherwise, an overflow event occurs. The bolded lines in Figure 5-2 show the admissible region for $N = 2$ in two dimensions.

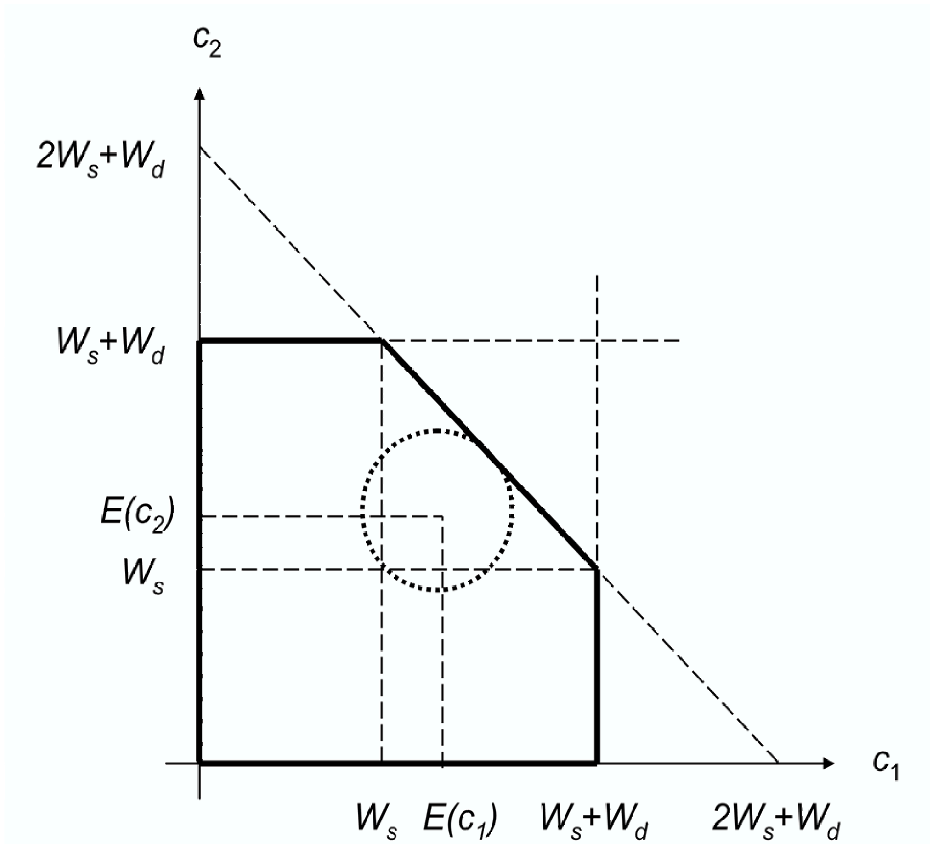


Figure 5-2: The admissible traffic region, in two dimensions, for $N = 2$. Three lines form the boundary constraints represented by (5.1). There are two lines each associated with a single element of the call vector $\mathbf{c}$, and one line associated with both elements of $\mathbf{c}$. The traffic sphere must be entirely contained within this admissible region for the link to be asymptotically non-blocking.

5.2.1 Asymptotic Analysis

We will consider the case where the number of users N becomes large, and use the law of large numbers to help us draw some conclusions. We can rewrite the call vector in the form

$$\mathbf{c} \;=\; \mu \cdot \mathbf{1} + \mathbf{c}'$$

where $\mathbf{1}$ is the length-N all-ones vector, and $\mathbf{c}' \sim \mathbf{N}(\mathbf{0}, \sigma^2\mathbf{1})$ is a zero-mean Gaussian random vector with i.i.d. components. Conceptually, we can visualize the random traffic vector as a random vector $\mathbf{c}'$ centered at $\mu\mathbf{1}$. The length of this random vector is given by

$$\|\mathbf{c}'\| \;=\; \sqrt{\sum_{n=1}^{N} c_i^2}$$

We will use an approach very similar to the sphere packing argument used in the proof of Shannon's channel capacity theorem in information theory [10]. We will show that asymptotically as the number of users becomes large, the traffic vector falls onto a sphere centered at the mean, and the provisioning becomes a problem of choosing the appropriate number of static and dynamic wavelengths so that this traffic sphere is completely contained within the admissible region.

From the law of large numbers, we know that

$$\frac{1}{N}\sum_{n=1}^{N} c_i^2 \rightarrow \sigma^2$$

as $N \rightarrow \infty$. This implies that asymptotically, as the number of users becomes large, the call vector $\mathbf{c}$ becomes concentrated on a sphere of radius $\sqrt{N}\sigma$ centered at the mean $\mu\mathbf{1}$. (This phenomenon is known as "sphere hardening" in the literature [11].)

Therefore, in order for the overflow probability to converge to zero, a necessary and sufficient condition is that the hyperplanes described by (5.1) enclose the sphere entirely. This is illustrated in Figure 5-2.

5.2.2 Minimum Distance Constraints

Next, we will derive necessary and sufficient conditions for the admissible traffic region to enclose the traffic sphere. Our goal is to ensure that we provision W_s and W_d such that the minimum distance from the center of the traffic sphere to the boundary of the admissible region is at least the radius of the sphere, therefore ensuring that all the traffic will fall within the admissible region.

Due to the identical distribution of the traffic for each user, the mean point $\mu\mathbf{1}$ will be equidistant from all planes whose description involves the same number of elements of $\mathbf{c}$. We define a *distance function* $f(n)$ such that $f(n)$ is the minimum distance from the mean $\mu\mathbf{1}$ to any hyperplane whose description involves n components of $\mathbf{c}$.

Lemma 15. *The distance function $f(n)$ from the traffic mean to a hyperplane involving n elements of the traffic vector $\mathbf{c}$ is given by*

$$f(n) = \sqrt{n}\left(W_s + \frac{W_d}{n} - \mu\right) \;, \quad n = 1, \ldots, N \tag{5.2}$$

Proof. The distance can be calculated using basic geometric principles and is essentially a simplified version of the proof of Lemma 16. □

We define the *minimum boundary distance* to be

$$F_{min} \;=\; \min_{n=1,\ldots,N} \; f(n)$$

A necessary and sufficient condition for the overflow probability to go to zero asymptotically with the number of users is

$$F_{min} \geq \sqrt{N}\sigma$$

We would like to determine the index n such that $f(n)$ is minimized. Unfortunately, this value of n turns out to depend on the choice of provisioning W_s. Let us consider the derivative of the distance function $f'(n)$:

$$\begin{aligned} f'(n) &= \frac{1}{2\sqrt{n}}\left(W_s + \frac{W_d}{n} - \mu\right) + \sqrt{n}\left(-\frac{W_d}{n^2}\right) \\ &= \frac{1}{2\sqrt{n}}\left(W_s - \frac{W_d}{n} - \mu\right) \end{aligned}$$

We can divide W_s into three regimes of interest, corresponding to different ranges of values for W_s and W_d, and characterize $f(n)$ in each of these regions:

Regime 1: If $W_s \leq \mu$:

In this region, $f'(n) < 0$ for all n. This implies that $f(n)$ is a decreasing function of n, and $F_{min} = f(N)$, giving a minimum distance of

$$F_{min} = \sqrt{N}\left(W_s + \frac{W_d}{N} - \mu\right)$$

Regime 2: If $\mu < W_s \leq \mu + W_d$:

In this region, $f'(n)$ starts out negative and ends up positive over $1 \leq n \leq N$. This implies that $f(n)$ is convex and has a minimum. Neglecting integrality concerns, this minimum occurs when $f'(n) = 0$, or

$$n^* = \frac{W_d}{W_s - \mu}$$

Therefore $F_{min} = f(n^*)$ in this regime. Substituting the appropriate values, it can be shown that the minimum distance is given by

$$F_{min} = 2\sqrt{W_d(W_s - \mu)}$$

Regime 3: If $W_s > \mu + W_d$:

In this region, $f'(n) > 0$ for all n. This implies that $f(n)$ is an increasing function of n, and $F_{min} = f(1)$, giving a minimum distance of

$$F_{min} = W_s + W_d - \mu$$

5.2.3 Optimal Provisioning

In the preceding section, we derived the minimum distance criteria for the hybrid system. Given a fixed number of statically allocated wavelengths W_s, we can use the equation $F_{min} \geq \sqrt{N}\sigma$ to calculate the minimum number of dynamic wavelengths W_d to achieve asymptotically non-overflow performance. We can also draw a few additional conclusions about provisioning hybrid systems.

Theorem 11. *A minimum of μ static wavelengths should always be provisioned per user.*

Proof. For $W_s \leq \mu$, we know from Case 1 above that the minimum distance constraint is

$$F_{min} = \sqrt{N}\left(W_s + \frac{W_d}{N} - \mu\right) \geq \sqrt{N}\sigma$$

$$W_s + \frac{W_d}{N} \geq \mu + \sigma$$

$$\Rightarrow W_{tot} = NW_s + W_d \geq (\mu + \sigma)N$$

Note that the total number of wavelengths $W_{tot} = NW_s + W_d$ is independent of W_s and W_d in this regime, suggesting that the same total number of wavelengths are required regardless of the partitioning between static and dynamic wavelengths. Since static wavelengths are less expensive to provision than dynamic wavelengths, this shows that there is never any reason to provision less than $W_s = \mu$ wavelengths. $\square$

An interesting corollary to this theorem follows from the observation that the case where $W_s = 0$ (i.e. all wavelengths are dynamic) also falls in this regime (i.e. Regime 1). Since fully dynamic provisioning is obviously the least-constrained version of this system, we can use it as a bound on the minimum number of wavelengths required by **any** asymptotically overflow-free system.

Corollary: For non-overflow operation, a lower bound on the number of wavelengths required is given by

$$W_{tot} \geq (\mu + \sigma)N$$

We can also consider a system that is fully static, with no dynamic provisioning. This is the most inflexible wavelength partitioning, and provides us with an upper bound on the number of wavelengths required by any hybrid system.

Theorem 12. *For a fully static system with no dynamic provisioning, the minimum number of wavelengths required is given by*

$$W_{tot} = (\mu + \sigma)N + \left(\sqrt{N} - 1\right) N\sigma$$

Proof. Let $W_d = 0$. Then, for overflow-free operation, we obviously need $W_s > \mu$. This puts us in Regime 3 where $W_s > \mu + W_d$, and the minimum distance condition gives us

$$\begin{aligned} F_{min} = W_s + W_d - \mu &> \sqrt{N}\sigma \\ W_s &> \mu + \sqrt{N}\sigma \\ &= \mu + \sigma + \left(\sqrt{N} - 1\right)\sigma \end{aligned}$$

$$W_{tot} = NW_s = (\mu + \sigma)N + \left(\sqrt{N} - 1\right) N\sigma$$

□

Note that this exceeds the lower bound on the minimum number of wavelengths by $(\sqrt{N} - 1)N\sigma$. We can therefore regard this quantity as the **maximum switching gain** that we can achieve in the hybrid system. This gain is measured in the maximum number of wavelengths that could be saved if all wavelengths were dynamically switched.

Combining the upper and lower bounds, we can make the following observation:

Corollary: For efficient overflow-free operation, the total number of wavelengths required by any hybrid system is bounded by

$$(\mu + \sigma)N \leq W_{tot} \leq (\mu + \sigma)N + (\sqrt{N} - 1)N\sigma$$

5.2.4 Simulations

Simulations were conducted to verify the accuracy of the provisioning results derived. Figure 5-3 verifies the results of the preceding discussion for the case of $\mu = 100$ and $\sigma = 10$. The rapidly descending curve shows that if the theoretical minimum of W_{tot} wavelengths are provisioned, then as the number of users N increases, the overflow probability drops off quickly and eventually the system becomes asymptotically non-blocking. The other two curves show that if less than W_{tot} wavelengths are provisioned, the overflow probability no longer converges to zero as the number of users increases.

Note also that the convergence occurs fairly rapidly if the W_{tot} wavelengths calculated in the preceding sections are provisioned. In a system with just 30 users, the overflow probability has already decreased to the order of 10^{-5}. Since the number of users is equal to the number of input-output fiber pairs, this corresponds to a link with as few as 5 input fibers and 6 output fibers, for example. Therefore, the results are useful in designing for good network performance even when N is finite and small.

5.3 Waveband-Granularity Switching

5.3.1 Asymptotic Analysis

In this section, we also consider a shared link, but now we allow for waveband switching. Again, each user is statically provisioned W_s wavelengths for its own use. Additionally, we assume there are b wavebands, each of size W_b. Each waveband can be assigned to serve calls from any user, but all W_b wavelengths within the same waveband must serve the same user.

This banded approach is interesting because, compared to wavelength switching, it allows for more wavelengths to be dynamically allocated, at the cost of coarser switching granularity. For example, given the same number of switches, a waveband approach with bands of $W_b = 2$ wavelengths will have twice the total number of dynamic wavelengths as a wavelength-switched network. However, these dynamic

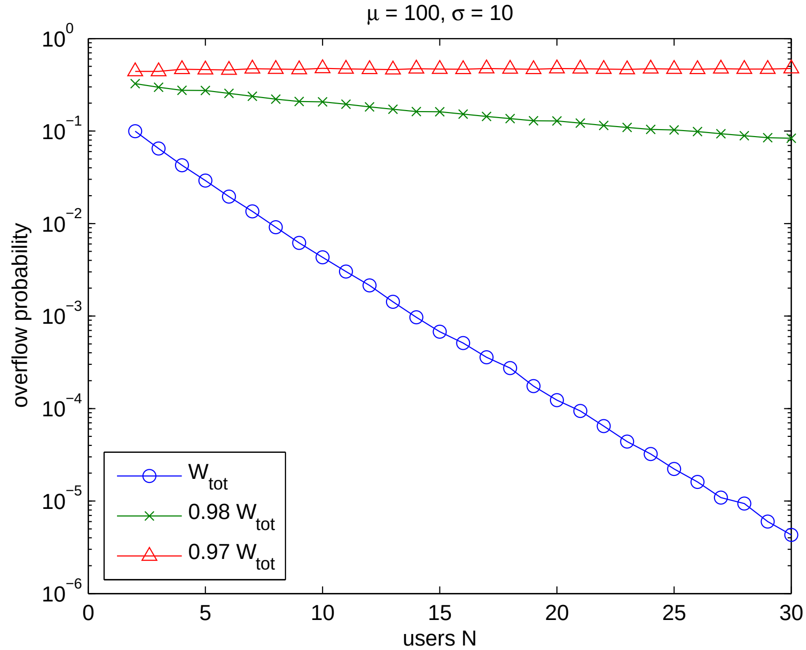


Figure 5-3: Curves show decrease in overflow probability with increasing number of users N. Note that if fewer than W_{tot} wavelengths are provisioned, the overflow probability no longer converges to zero as the number of users increases.

wavelengths are less flexible in the banded case, since two dynamic wavelengths must be assigned to a user at a time.

Note that the wavelength-switched network is a special case of the waveband-switched network where $W_b = 1$. In such a case, $W_d = b \cdot W_b = b$. In this section, we will analyze the performance of banded networks in general, and make some comparisons about the performance improvement gained over wavelength-switched networks by allowing waveband switching.

Again, we assume that the traffic vector $\mathbf{c}$ is composed of normally distributed i.i.d. entries with mean μ and variance σ^2. An overflow event occurs if there exists at least one call that is blocked due to insufficient wavelengths being available to service it. This can be written mathematically as

$$\sum_{i=1}^{N} \max\left\{ \left\lceil \frac{c_i - W_s}{W_b} \right\rceil \ , \ 0 \right\} > b \tag{5.3}$$

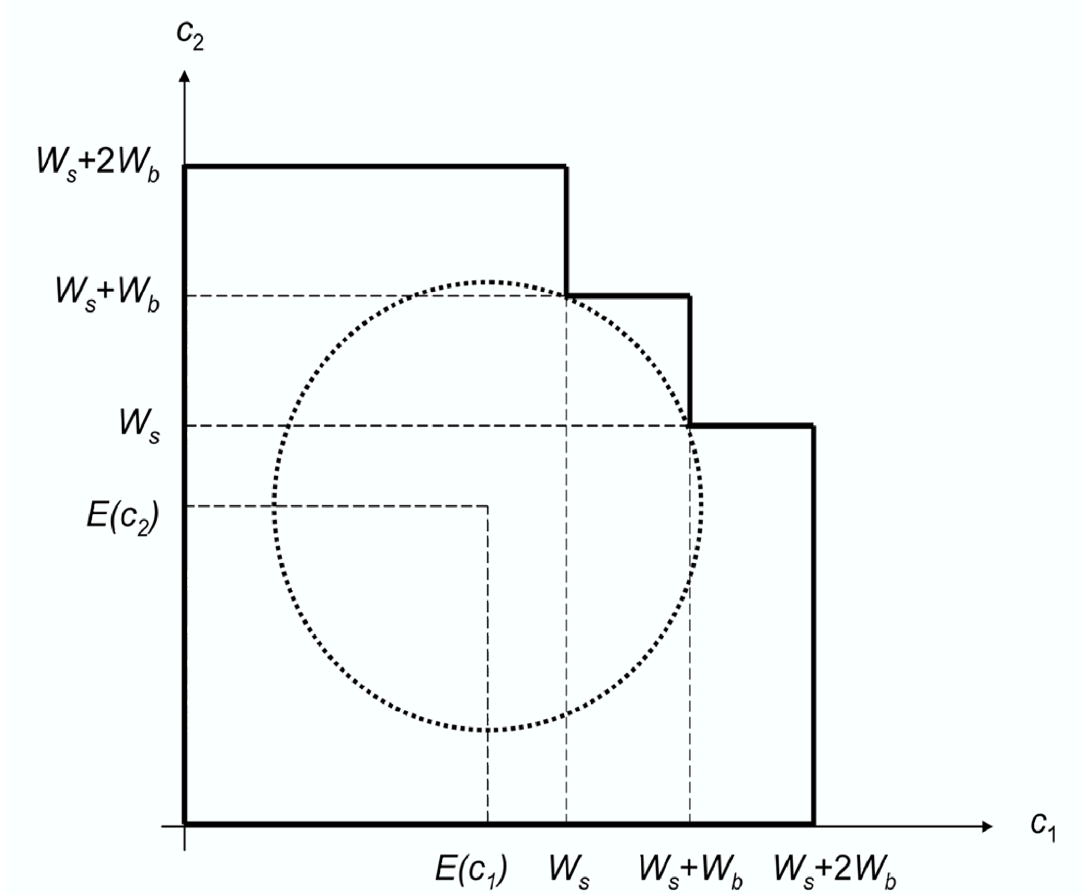


Figure 5-4: The admissible region for a link with $N = 2$ users. The traffic sphere must be entirely enclosed within the admissible region in order for the link to be asymptotically non-blocking.

(5.3) can be written as N sets of boundary constraints, where the n^{th} set consists of $C(N,n) = \frac{N!}{(N-i)!\ n!}$ equations, each involving n elements of the traffic vector $\mathbf{c}$:

$$\left\lceil \frac{c_i - W_s}{W_b} \right\rceil \leq b \quad , \quad \forall\, i$$

$$\left\lceil \frac{c_i - W_s}{W_b} \right\rceil + \left\lceil \frac{c_j - W_s}{W_b} \right\rceil \leq b \quad , \quad \forall\, i \neq j$$

$$\vdots$$

Figure 5-4 illustrates an example of the admissible traffic region for $N = 2$ users.

We will next determine the distance from the mean of the traffic vector to each boundary, and derive a sufficient mathematical condition for the traffic vector to be admissible (i.e. not in overflow). Consider the n^{th} set of boundary constraints, and suppose that the first n elements of $\mathbf{c}$ are active. Then we require

$$\sum_{i=1}^{n} \left\lceil \frac{c_i - W_s}{W_b} \right\rceil \quad \leq \quad b \tag{5.4}$$

We observe that since all the terms in the summation are integers, and b is an integer, (5.4) holds if and only if

$$\frac{c_1 - W_s}{W_b} + \sum_{i=2}^{n} \left\lceil \frac{c_i - W_s}{W_b} \right\rceil \quad \leq \quad b \tag{5.5}$$

Therefore we can equivalently consider provisioning W_s and W_b to satisfy (5.5), and (5.4) will follow. Using the inequality $\lceil x \rceil < x + 1$, we observe that

$$\sum_{i=2}^{n} \left\lceil \frac{c_i - W_s}{W_b} \right\rceil \leq \sum_{i=2}^{n} \left(\frac{c_i - W_s}{W_b} + 1 \right)$$

from which we can conclude that (5.5) holds if we choose W_s and W_b such that

$$\frac{c_1 - W_s}{W_b} + \sum_{i=2}^{n} \left(\frac{c_i - W_s}{W_b} + 1 \right) \quad \leq \quad b \tag{5.6}$$

This is equivalent to

$$\frac{c_1 - W_s}{W_b} + \sum_{i=2}^{n} \frac{c_i - W_s}{W_b} \quad \leq \quad b - (n-1)$$

Rearranging the above, we obtain:

$$\sum_{i=1}^{n} c_i \leq nW_s + (b - (n-1))\, W_b \tag{5.7}$$

By the above reasoning, we have that (5.7) → (5.6) → (5.5) → (5.4). Therefore (5.7) is a sufficient condition for the traffic vector being admissible.

Recall that we derived this expression assuming that the first n elements of $\mathbf{c}$ were active. In general, for n active elements, the sum on the left of (5.7) will involve the sum of those active elements. We also point out here that by using the upper bound $\lceil x \rceil < x + 1$ as the basis for our provisioning, we have been more conservative in our estimate of the traffic. We therefore expect that this will result in a small amount of overprovisioning of the link.

5.3.2 Minimum Distance Constraints

We define the minimum distance from the traffic mean to any boundary involving n active constraints to be $f(n)$. This minimum distance expression will later be useful in determining sufficient provisioning for overflow-free operation.

Lemma 16. *The distance $f(n)$ from the traffic mean to any hyperplane involving n elements of the traffic vector $\mathbf{c}$ is given by:*

$$f(n) = \sqrt{n}\left(W_s + \frac{b-(n-1)}{n} W_b - \mu\right) \tag{5.8}$$

Proof. This is essentially a basic geometric exercise. For a fixed n, the hyperplane has a normal vector consisting of n unity entries and $N-n$ zero entries. Since by symmetry the mean of the traffic is equidistant from all hyperplanes with the same number of active constraints, without loss of generality, assume that the first n constraints that are active. Then the closest point on the hyperplane has the form

$$[\mu + x, \ldots, \mu + x, \mu, \ldots, \mu]$$

where the first n entries are $\mu+x$, and the remainder are μ. The value of x is obtained applying (5.7), which requires that

$$\begin{aligned}
\sum_{i=1}^{n}(\mu + x) &= nW_s + (b - (n-1))W_b \\
\Rightarrow nx &= nW_s + (b - (n-1))W_b - n\mu \\
x &= W_s + \frac{b-(n-1)}{n}W_b - \mu
\end{aligned}$$

The distance from the point $[\mu, \ldots, \mu]$ to this point on the hyperplane is

$$\| \; [\mu + x, \ldots, \mu + x, \mu, \ldots, \mu] - [\mu, \ldots, \mu] \; \|$$

$$= \sqrt{nx^2}$$
$$= \sqrt{n}\; x$$

where, after substituting for x, we obtain

$$f(n) = \sqrt{n}\left(W_s + \frac{b-(n-1)}{n}W_b - \mu\right)$$

□

By the same law of large numbers reasoning as in Section 5.2.1, we observe that asymptotically as the number of users N becomes large, the traffic will converge to a sphere of radius $\sqrt{N}\sigma$ centered at the mean. Let the minimum distance from the mean to the closest hyperplanar boundary be $F_{min} = \min_n f(n)$. Then the system will be asymptotically non-blocking if $F_{min} > \sqrt{N}\sigma$.

In the preceding wavelength-switched networks, fixing the number of switches b was equivalent to fixing W_d. This left only 1 free parameter, W_s, and uniquely determined the optimal provisioning for asymptotically non-blocking operation. However, in this section, if b is fixed, *two* parameters remain to be chosen: the number of static wavelengths W_s and the waveband size W_b. Therefore, the choice of these parameters is not unique unless we specify additional optimization criteria.

If in addition we require that the optimal provisioning also minimize the total number of wavelengths $W_{tot} = NW_s + bW_b$ for a fixed b, there will exist a unique choice of W_s and W_b for each b. We now proceed to derive this optimal choice.

We would like to first determine the index n that minimizes $f(n)$. We consider the first derivative of $f(n)$:

$$\begin{aligned} f'(n) &= \frac{1}{2\sqrt{n}}\left[W_s + \left(\frac{b+1}{n} - 1\right) W_b - \mu\right] \\ &\quad - \sqrt{n}\left(\frac{b+1}{n^2}\right) \\ &= \frac{1}{2\sqrt{n}}\left[W_s - \left(\frac{b+1}{n} + 1\right) W_b - \mu\right] \end{aligned} \tag{5.9}$$

From (5.9), we observe that the behavior of $f(n)$ can be characterized in three regimes corresponding to different ranges of values for W_s and W_b. By analysis very similar to Section 5.2.2 (see Section 5.7), the minimum number of wavelengths required in each regime, along with the choice of W_s and W_b that achieves this, is listed in Table 5.1.

Table 5.1: Wavelength Requirements for Asymptotically Non-blocking Waveband-Switched Networks

Regime 1: $W_s \le \mu + \left(\frac{b+1}{N} + 1\right) W_b$

b	W_{tot}	W_s	W_b
$b \le \frac{N\sigma}{2} - 1$	$N\mu + N\sigma + \left[\frac{N(N-1)}{2(b+1)}\right]\sigma$	$\mu + \frac{1}{2}\left(1 + \frac{N}{b+1}\right)\sigma$	$\frac{1}{2}\left(\frac{N}{b+1}\right)\sigma$
$\frac{N\sigma}{2} - 1 < b \le N(\mu+\sigma+1) - 1$	$N\left[\mu + \sigma + \frac{N-1}{N}\right]$	$\mu + \sigma + 1 - \frac{b+1}{N}$	1
$b > N(\mu+\sigma+1) - 1$	b	0	1

Regime 2: $\mu + \left(\frac{b+1}{N} + 1\right) W_b < W_s \le \mu + (b+2)W_b$

b	W_{tot}	W_s	W_b
$b \le N\sigma$	$N\mu + N\left[\frac{b+N}{\sqrt{(b+1)(b+N)}}\right]\sigma$	$\mu + \left[\frac{(b+2N)}{2\sqrt{(b+1)(b+N)}}\right]\sigma$	$\left[\frac{N}{2\sqrt{(b+1)(b+N)}}\right]\sigma$
$b > N\sigma$	$N\left(\mu + \frac{N}{4(b+1)}\ \sigma^2\right) + b$	$\mu + \frac{N}{4(b+1)}\ \sigma^2$	1

Regime 3: $W_s > \mu + (b+2)W_b$

b	W_{tot}	W_s	W_b
$b \le \frac{\sqrt{N}}{2}\sigma - 1$	$N\left(\mu + \frac{b+2}{b+1}\frac{\sqrt{N}}{2}\ \sigma\right) + \frac{b\sqrt{N}}{2(b+1)}\sigma$	$\mu + \frac{b+2}{b+1}\frac{\sqrt{N}}{2}\ \sigma$	$\frac{1}{b+1}\frac{\sqrt{N}}{2}\ \sigma$
$b > \frac{\sqrt{N}}{2}\sigma - 1$	$N\left(\mu + \sqrt{N}\sigma\right)$	$\mu + \sqrt{N}\sigma$	0

5.3.3 Optimal Provisioning

For a given number of switches b, Section 5.3.2 gives the total number of wavelengths in each of three regimes. To determine the optimal operating regime, these expressions should be evaluated for the particular values of b, μ, and σ, and the regime that requires the fewest total wavelengths W_{tot} should be chosen as the operating regime. The results in Section 5.3.2 for that regime will then provide the optimal choice of W_s and W_b to achieve asymptotically non-blocking operation.

Figure 5-5 plots the dropoff in overflow probability as the number of users N increases. The different curves show the effect of underprovisioning the total number of wavelengths relative to the predicted W_{tot}. As expected, W_{tot} slightly overestimates the total number of wavelengths required, but not by much. To see this, observe that if only $0.94W_{tot}$ is provisioned (the curve with the square points), the overflow probability stays constant at a high value even as the number of users increases and does not diminish.

Figure 5-6 shows an example of the number of wavelengths per user required as a function of the number of switches per user for a 1000-user network. Recall from Theorem 11 that a strict lower bound on the number of wavelengths required is $\mu + \sigma$ wavelengths per user, achieved using full switching requiring $\mu + \sigma$ switches per user. From the figure, we observe that with a relatively small number of switches per user, we can come close to this lower bound – that is, the minimum number of $\mu + \sigma = 110$ wavelengths can be approached very closely with only 3 to 5 switches per user instead of 110 switches per user.

5.3.4 Typical Operating Regimes

In this section, we seek a closed-form, approximate expression for the total number of wavelengths required W_{tot} and the static and dynamic provisioning W_s and W_b that applies in typical network operating regimes. This will give us an intuitive sense of the relationship between these quantities and the network parameters such as the traffic mean μ and variance σ^2 and the number of switches b.

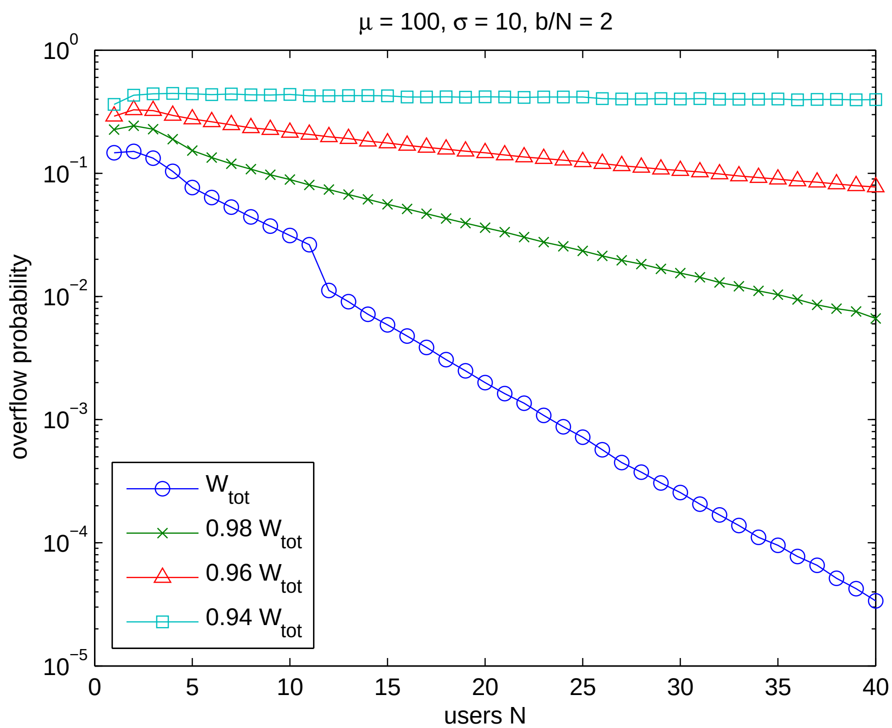


Figure 5-5: Dropoff in overflow probability as the number of users N increases. In this example, two switches per user are available. The different curves show the effect of underprovisioning the total number of wavelengths relative to the theoretical minimum. Note that if less than 94% of the calculated wavelengths are provisioned, the overflow probability does not decrease even as the number of users increases.

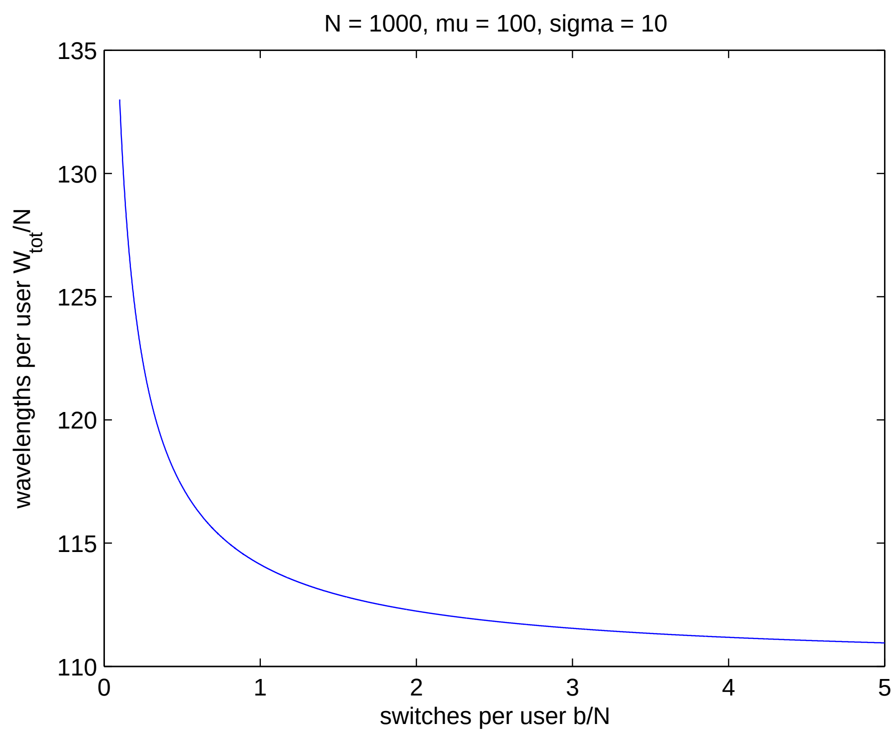


Figure 5-6: A plot of the number of wavelengths required as a function of the number of switches. Note that the initial wavelength savings is significant, but the marginal gain in wavelength savings decreases rapidly as the number of switches gets larger. Theorem 11 gives a lower bound on the number of wavelengths required as $\mu + \sigma = 110$ wavelengths per user in this example.

We will assume that in a typical environment, $b/N \leq \sigma/2$. From Figure 5-6, we see that this is a reasonable assumption, since we can approach the minimum number of wavelengths required with a relatively small number of switches per user. We also assume that N is large. We next consider the three regimes to determine which regime is optimal under these assumptions.

In Regime 1, we have $b/N \leq \sigma/2$, and the total number of wavelengths required is

$$\begin{aligned} W_{tot}^{(1)} &= N\mu + N\sigma + \left[\frac{N(N-1)}{2(b+1)}\right]\sigma \\ &\approx N\mu + N\left[1 + \frac{N^2}{2b}\right]\sigma \end{aligned}$$

In Regime 2, it follows from our assumptions that $b/N < \sigma$ and we have

$$\begin{aligned} W_{tot}^{(2)} &= N\mu + N\left[\frac{b+N}{\sqrt{(b+1)(b+N)}}\right]\sigma \\ &\approx N\mu + N\left[\sqrt{\frac{b+N}{b}}\right]\sigma \\ &= N\mu + N\left[\sqrt{1+\frac{N}{b}}\right]\sigma \end{aligned}$$

Since $\frac{1}{2\sqrt{N}}$ approaches zero as N increases, for sufficiently large N we must have $b/N > \frac{1}{2\sqrt{N}}\sigma$ in Regime 3 and

$$W_{tot}^{(3)} = N\mu + N\sqrt{N}\sigma$$

By inspection, we observe that the minimum number of wavelengths is achieved in Regime 2:

$$W_{tot} \approx N\mu + N\left[\sqrt{1+\frac{N}{b}}\right]\sigma$$

with

$$\begin{aligned} W_s &\approx \mu + \frac{b+2N}{2\sqrt{b(b+N)}}\,\sigma \\ W_b &\approx \frac{N}{2\sqrt{b(b+N)}}\,\sigma \end{aligned}$$

Comparing the total number of static wavelengths NW_s with the total number of dynamic wavelengths bW_b, we observe that

$$NW_s \approx N\mu + \frac{N^2}{\sqrt{b(b+N)}} + bW_b$$

which gives a sense of the optimal relative amounts of static and dynamic provisioning that is appropriate. Note that we can take advantage of the predictability of the traffic to provision significantly more wavelengths statically.

5.4 Provisioning for Small Numbers of Users

The analysis in Sections 5.2 and 5.3 is asymptotic in the number of users. Recall that each "user" in our shared-link context corresponds to an input-output fiber pair. In practical networks, we rarely have an infinite number of such users. However, we also rarely need to have strictly non-blocking networks – typically, we will have a target overflow probability that we consider to be sufficiently low to deliver good service. We would therefore also like to know how to use the results from our asymptotic approach in the preceding sections to networks with a small number of users in which we may allow a target overflow probability. In this section, we discuss how to adapt our results to this scenario.

5.4.1 Statistics of the Traffic Vector Length

Consider the distance R of the traffic vector $\mathbf{c}$ from its mean point $\mu\mathbf{1}$:

$$R = \|\mathbf{c} - \mu\mathbf{1}\| \quad = \quad \sqrt{\sum_{n=1}^{N}(c_i - \mu)^2}$$

Define a new quantity $X = R^2$. For a fixed N, X is a random variable with mean and variance given by

$$\begin{aligned} E[X] &= E\left[\sum_{n=1}^{N}(c_i - \mu)^2\right] \\ &= \sum_{n=1}^{N} E[(c_i - \mu)^2] \\ &= \sum_{n=1}^{N} \sigma^2 = N\sigma^2 \end{aligned}$$

$$\begin{aligned} \mathsf{var}(X) &= \mathsf{var}\left[\sum_{n=1}^{N}(c_i - \mu)^2\right] \\ &= \sum_{n=1}^{N} \mathsf{var}\left[(c_i - \mu)^2\right] \\ &= \sum_{n=1}^{N} \left(E\left[(c_i - \mu)^4\right] - E\left[(c_i - \mu)^2\right]^2\right) \\ &= \sum_{n=1}^{N} \left(3\sigma^4 - \sigma^4\right) = 2N\sigma^4 \end{aligned}$$

where the central moments of a Gaussian random variable can be found in most probability texts (e.g. [25]).

Note that X is the sum of N squared Gaussian random variables and by the

Central Limit Theorem can itself be approximated by a Gaussian random variable with mean $N\sigma^2$ and standard deviation $\sqrt{2N}\sigma^2$. We can observe that the standard deviation of X as a fraction of the mean decreases with increasing N as

$$\frac{\sqrt{2N}\sigma^2}{N\sigma^2} = \sqrt{\frac{2}{N}}$$

5.4.2 Practical Network Provisioning

In practical network provisioning, it is often not necessary to achieve totally non-blocking operation – it suffices if the blocking probability is sufficiently low. Suppose Q is the target overflow probability for the network (i.e. we wish to design the network so that the probability of overflow is at most Q).

The probability that the traffic vector falls within a sphere of radius r centered at the mean is $P(R \leq r)$. We choose r to satisfy

$$\begin{aligned} P(R \leq r) &= Q \\ \Rightarrow P(X \leq r^2) &= Q \\ P\left(\frac{X}{\sigma^2} \leq \frac{r^2}{\sigma^2}\right) &= Q \\ \Rightarrow P\left(Y \leq \frac{r^2}{\sigma^2}\right) &= Q \end{aligned}$$

where Y is defined as X/σ^2. Observe that since Y is the sum of N zero-mean unit variance Gaussian random variables, it is itself a chi-squared random variable with N degrees of freedom. Then:

$$r = \sigma \sqrt{\mathsf{chi2inv}(Q, N)}$$

where $\mathsf{chi2inv}(Q, N)$ is the inverse CDF of a chi-squared random variable with N

degrees of freedom at the point Q.

Since r is now the radius of a traffic sphere within which the realized traffic vector will be found with desired probability Q, the wavelength provisioning should be chosen so that the minimum distance from the traffic mean to the nearest boundary constraint hyperplane F_{min} exceeds r. This will ensure that an overflow event happens with probability no greater than Q.

5.4.3 Numerical Example

Recall that Table 5.1 was calculated assuming that the mean point needed to be a minimum distance of $\sqrt{N}\sigma$ from all boundary hyperplanes, while for finite N we now require the minimum distance to be at least r. We can therefore conclude that the expressions for each regime in Table 5.1 hold for finite N also, as long as σ in the table is replaced with $r/\sqrt{N}$.

Figure 5-7 shows the total number of wavelengths required for a link with $\mu = 100$, $\sigma = 10$, and 2 switches per user for both a 1% overflow probability and the asymptotic minimum W_{tot}. Note that even for a small number of users, the number of extra wavelengths required (compared to the asymptotic minimum) is not large, and diminishes rapidly as the number of users increases.

5.5 Non-IID Traffic

The majority of this chapter has dealt with the case of independent identically-distributed user traffic: we have assumed that $\mu_i = \mu$ and $\sigma_i^2 = \sigma^2$ for all users i. In many scenarios this will not be the case. Some input-output fiber pairs may connect more active sections of the network than others, leading to users with differing traffic characteristics. In this section, we discuss how to deal with non-IID traffic scenarios.

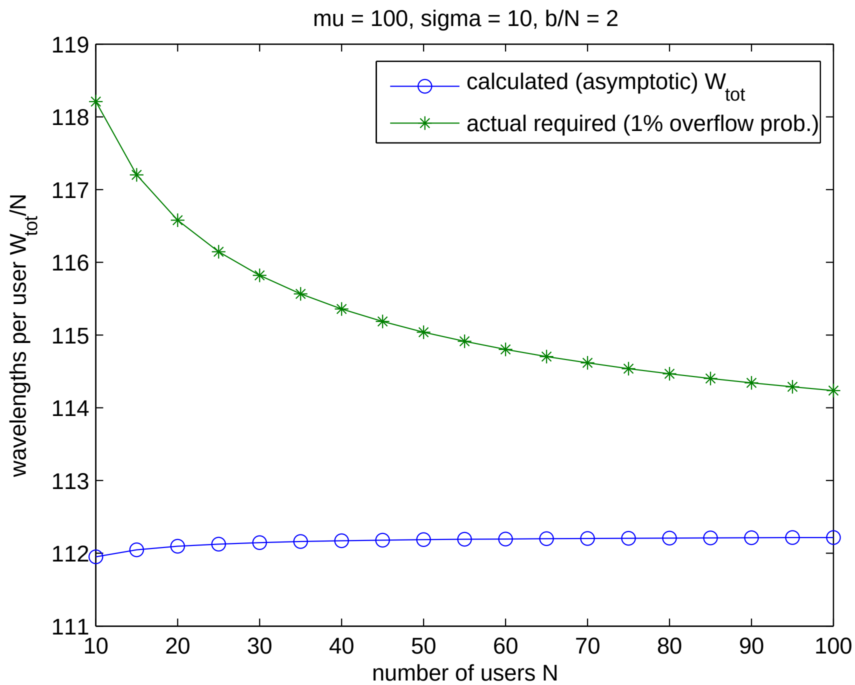


Figure 5-7: A shared link provisioned for 1% overflow probability. The total number of wavelengths required for a link with $\mu = 100$, $\sigma = 10$, and 2 switches per user is shown both for 1% overflow probability and the asymptotic minimum W_{tot}.

5.5.1 Asymptotic Analysis

We consider the waveband-granularity shared-link model of Section 5.3, with one exception: each user i is now characterized by traffic c_i, where $c_i \sim \mathsf{N}(\mu_i, \sigma_i^2)$. It now makes sense to allow for a different number of static wavelengths $W_s^{(i)}$ to be provisioned per user. As before, an overflow occurs if

$$\sum_{i=1}^{N} \max\left\{ \left\lceil \frac{c_i - W_s^{(i)}}{W_b} \right\rceil \ , \ 0 \right\} > b \tag{5.10}$$

We next define a set of new random variables $\hat{c}_i$, where

$$\hat{c}_i \;=\; \frac{c_i - \mu_i}{\sigma_i}$$

Note that each $\hat{c}_i$ is now an IID standard Gaussian random variable with mean 0 and variance 1. We can rewrite (5.10) in the form

$$\sum_{i=1}^{N} \max\left\{ \left\lceil \frac{\sigma_i \hat{c}_i + \mu_i - W_s^{(i)}}{W_b} \right\rceil \ , \ 0 \right\} > b \tag{5.11}$$

Following the approach of Section 5.3, consider the n^{th} set of boundary constraints, and suppose that the first n elements of $\mathbf{c}$ are active. Then we require

$$\sum_{i=1}^{n} \left\lceil \frac{\sigma_i \hat{c}_i + \mu_i - W_s^{(i)}}{W_b} \right\rceil \ \leq \ b \tag{5.12}$$

We observe that since all the terms in the summation are integers, and b is an integer, (5.12) holds if and only if

$$\frac{\sigma_1 \hat{c}_1 + \mu_1 - W_s^{(1)}}{W_b} + \sum_{i=2}^{n} \left\lceil \frac{\sigma_i \hat{c}_i + \mu_i - W_s^{(i)}}{W_b} \right\rceil \quad \leq \quad b \tag{5.13}$$

Therefore we can equivalently consider provisioning $W_s^{(i)}$ and W_b to satisfy (5.13), and (5.12) will follow. Using the inequality $\lceil x \rceil < x + 1$, we can conclude that (5.13) holds if we choose W_s and W_b such that

$$\frac{\sigma_1 \hat{c}_1 + \mu_i - W_s^{(1)}}{W_b} + \sum_{i=2}^{n} \left(\frac{\sigma_i \hat{c}_i + \mu_i - W_s^{(i)}}{W_b} + 1 \right) \quad \leq \quad b \tag{5.14}$$

This is equivalent to

$$\frac{\sigma_1 \hat{c}_1 + \mu_1 - W_s^{(1)}}{W_b} + \sum_{i=2}^{n} \frac{\sigma_i \hat{c}_i + \mu_i - W_s^{(i)}}{W_b} \quad \leq \quad b - (n-1)$$

Rearranging the above, we obtain:

$$\sum_{i=1}^{n} \sigma_i \hat{c}_i \quad \leq \quad (b - (n-1))\, W_b + \sum_{i=1}^{n} W_s^{(i)} \tag{5.15}$$

By the above reasoning, we have that (5.15) → (5.14) → (5.13) → (5.12). Therefore (5.15) is a sufficient condition for the traffic vector being admissible.

Note that the equations in (5.15) again describe sets of hyperplanes that form the admissible region for the traffic vector $\hat{\mathbf{c}} = [\hat{c}_1, \ldots, \hat{c}_N]$. As the number of users becomes large, the traffic vector will concentrate itself on a sphere of radius $\sqrt{N}$ centered at the origin. Therefore, a necessary and sufficient condition for the system to be asymptotically non-blocking is simply for the minimum distance from the origin to each of the hyperplanes to be at least $\sqrt{N}$.

5.6 Chapter Summary

In this chapter, we examined wavelength provisioning for a shared link in a backbone network. We considered networks with both static and dynamically provisioned wavelengths. We argued that for networks where the traffic has a predictable behavior, efficient networks with reduced complexity could be achieved by statically provisioning wavelengths to support much of the traffic since it has an average behavior, while provisioning additional shared dynamic wavelengths to accommodate the randomness around this average.

Using a geometric argument, we obtained asymptotic results for the optimal wavelength provisioning on the shared link. We proved that the number of static wavelengths should be sufficient to support at least the traffic mean. We derived in closed form expressions for the optimal provisioning of the shared link given the mean μ and variance σ^2 of the traffic.

We also showed that by allowing the dynamic wavelengths to be switched in bands of multiple wavelengths rather than individually, very efficient networks can be achieved while using a very small number of switches per user. We again derive the optimal static and dynamic provisioning as well as the optimal waveband size given the traffic characterization. Finally, we showed that the results could be adapted for networks with a small, finite number of users with a fixed target overflow probability. A method for extending the approach to networks with asymmetric user traffic was also given.

5.7 Chapter Appendix

In this appendix, we will calculate the minimum number of wavelengths required to achieve asymptotically non-blocking performance in each of the three regimes of $f(n)$.

Recall that

$$
\begin{aligned}
f(n) &= \sqrt{n}\left[W_s + \left(\frac{b+1}{n} - 1\right) W_b - \mu\right] \\
f'(n) &= \frac{1}{2\sqrt{n}}\left[W_s - \left(\frac{b+1}{n} + 1\right) W_b - \mu\right]
\end{aligned}
$$

Regime 1: $W_s \leq \mu + \left(\frac{b+1}{N} + 1\right) W_b$

In this regime, $f'(n) \leq 0$ for all n, so $f(n)$ is monotonically decreasing and $F_{min} = f(N)$. To satisfy the minimum distance constraint, we need

$$
f(N) = \sqrt{N}\left[W_s + \left(\frac{b+1}{N} - 1\right) W_b - \mu\right] \quad > \quad \sqrt{N}\sigma
$$

which can be reduced to

$$
W_{tot} = NW_s + bW_b \quad > \quad N(\mu + \sigma) + (N-1)W_b
$$

Note that the expression on the right is independent of W_s. Therefore, we can minimize W_{tot} by minimizing W_b (or equivalently, maximizing W_s) within this regime. Since in this regime $W_s \leq \mu + \left(\frac{b+1}{N} + 1\right) W_b$, the maximum value of W_s is

$$
W_s = \mu + \left(\frac{b+1}{N} + 1\right) W_b
$$

The minimum distance then becomes:

$$
\begin{aligned}
f(N) &= \sqrt{N}\left[W_s + \left(\frac{b+1}{N} - 1\right) W_b - \mu\right] \\
&= \sqrt{N}\left[\left(\frac{b+1}{N} + 1\right) W_b + \left(\frac{b+1}{N} - 1\right) W_b\right] \\
&= 2\sqrt{N}\left(\frac{b+1}{N}\right) W_b
\end{aligned}
$$

We need $f(N) \geq \sqrt{N}\sigma$. The minimum necessary amount of provisioning occurs at equality:

$$
\begin{aligned}
2\sqrt{N}\left(\frac{b+1}{N}\right) W_b &= \sqrt{N}\sigma \\
\Rightarrow W_b &= \frac{1}{2}\left(\frac{N}{b+1}\right)\sigma
\end{aligned}
$$

The corresponding static provisioning can be calculated to be

$$
\begin{aligned}
W_s &= \mu + \left(\frac{b+1}{N} + 1\right) W_b \\
&= \mu + \frac{1}{2}\left(1 + \frac{N}{b+1}\right)\sigma
\end{aligned}
$$

Note that we have neglected integer issues in our calculations. Due to integrality, W_b is constrained to be at least 1 wavelength. Therefore, if $\frac{1}{2}\left(\frac{N}{b+1}\right)\sigma < 1$, then we will choose $W_b = 1$. This occurs when $b > \frac{N\sigma}{2} - 1$.

First suppose that $b \leq \frac{N\sigma}{2} - 1$. Then the total number of wavelengths is

$$\begin{aligned}
W_{tot} &= NW_s + bW_b \\
&= N\left[\mu + \frac{1}{2}\left(1 + \frac{N}{b+1}\right)\sigma\right] + b\left[\frac{1}{2}\left(\frac{N}{b+1}\right)\sigma\right] \\
&= N\mu + N\sigma + \left[\frac{N(N-1)}{2(b+1)}\right]\sigma
\end{aligned}$$

Next consider when $b > \frac{N\sigma}{2} - 1$. Then we choose $W_b = 1$ and W_s can be reduced and still satisfy the minimum distance constraint:

$$\begin{aligned}
f(N) &= \sqrt{N}\left[W_s + \left(\frac{b+1}{N} - 1\right)\cdot 1 - \mu\right] > \sqrt{N}\sigma \\
\Rightarrow W_s &= \mu + \sigma + 1 - \frac{b+1}{N}
\end{aligned}$$

Then the total number of wavelengths can be shown to be

$$\begin{aligned}
W_{tot} &= NW_s + bW_b \\
&= N\left[\mu + \sigma + 1 - \frac{b+1}{N}\right] + b\cdot 1 \\
&= N\left[\mu + \sigma + \frac{N-1}{N}\right]
\end{aligned}$$

Note that we must still have $W_s \geq 0$, so if $\mu + \sigma + 1 - \frac{b+1}{N} < 0$, then we must choose $s = 0$. This occurs when $b > N(\mu + \sigma + 1) - 1$. Then $W_{tot} = b$.

Regime 2: $\mu + \left(\frac{b+1}{N} + 1\right) W_b < W_s \leq \mu + (b+2)W_b$

In this regime, $f'(n)$ is initially negative but becomes positive as n increases. Therefore $f(n)$ is convex, and a minimum exists at the point where $f'(n) = 0$. Neglecting integrality constraints, we have

$$\frac{1}{2\sqrt{n^*}}\left[W_s - \left(\frac{b+1}{n^*}+1\right)W_b - \mu\right] = 0$$

$$\Rightarrow n^* = \frac{(b+1)W_b}{W_s - W_b - \mu}$$

The minimum distance is therefore given by

$$\begin{aligned} f(n^*) &= \sqrt{n^*}\left[W_s + \left(\frac{b+1}{n^*}-1\right)W_b - \mu\right] \\ &= 2\sqrt{(b+1)W_b(W_s - W_b - \mu)} \end{aligned}$$

and for asymptotically non-blocking operation we require that $f(n^*) \geq \sqrt{N}\sigma$.

We can now summarize the objective. We wish to minimize the total number of wavelengths

$$W_{tot} = NW_s + bW_b$$

subject to

$$2\sqrt{(b+1)W_b(W_s - W_b - \mu)} > \sqrt{N}\sigma$$

$$\mu < s < \mu + (b+1)W_b$$

We can solve this constrained optimization using the method of Lagrange multipliers:

$$
\begin{aligned}
L(W_s, W_b, \lambda) &= NW_s + bW_b \\
&\quad -\lambda[4(b+1)W_b(W_s - W_b - \mu) \\
&\quad -N\sigma^2] \\
\frac{\delta L}{\delta W_s} &= N - 4\lambda(b+1)W_b = 0 \qquad (5.16)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\delta L}{\delta W_b} &= b - [4\lambda(b+1)(W_s - W_b - \mu) \\
&\quad -4\lambda(b+1)W_b]
\end{aligned}
$$

$$
\begin{aligned}
&= b - 4\lambda(b+1)(W_s - \mu - 2W_b) \\
&= 0 \qquad (5.17)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\delta L}{\delta \lambda} &= 4(b+1)W_b(W_s - W_b - \mu) \\
&\quad -N\sigma^2 = 0 \qquad (5.18)
\end{aligned}
$$

From (5.16) and (5.17),

$$
W_b = \frac{N}{4\lambda(b+1)} \qquad (5.19)
$$

$$
\begin{aligned}
W_s &= \frac{b}{4\lambda(b+1)} + \mu + 2W_b \\
&= \mu + \frac{b+2N}{4\lambda(b+1)} \qquad (5.20)
\end{aligned}
$$

Using these results, we derive the following useful expressions:

$$W_s - W_b - \mu = \frac{b+N}{4\lambda(b+1)}$$

$$4(b+1)W_b(W_s - W_b - \mu) = \frac{N(b+N)}{4\lambda^2(b+1)}$$

Substituting into (5.18),

$$\frac{N(b+N)}{4\lambda^2(b+1)} - N\sigma^2 = 0$$

$$\frac{b+N}{4\lambda^2(b+1)} = \sigma^2$$

$$\Rightarrow \quad \lambda = \frac{1}{2\sigma}\sqrt{\frac{b+N}{b+1}}$$

Substituting back into (5.19) and (5.20),

$$\begin{aligned} W_s &= \mu + \frac{b+2N}{4(b+1)} \cdot 2\sigma\sqrt{\frac{b+1}{b+N}} \\ &= \mu + \left[\frac{(b+2N)}{2\sqrt{(b+1)(b+N)}}\right]\sigma \end{aligned}$$

$$\begin{aligned} W_b &= \frac{N}{4(b+1)} \cdot 2\sigma\sqrt{\frac{b+1}{b+N}} \\ &= \frac{N}{2\sqrt{(b+1)(b+N)}}\,\sigma \end{aligned}$$

As in the other regimes, we note that if b becomes sufficiently large, then the above expression will be less than 1, and we must choose either $W_b = 1$ or $W_b = 0$.

If $W_b = 0$, then the minimum distance criterion cannot be satisfied in this regime. Therefore our only choice is $W_b = 1$, which yields

$$W_s = \mu + \frac{N}{4(b+1)}\,\sigma^2$$

If we assume that "sufficiently large" occurs at $b \gg N$, then $W_b \approx N\sigma/2b$, and we choose $W_b = 1$ when $b > N\sigma/2$.

The total number of wavelengths required is therefore

$$\begin{aligned}
W_{tot} &= NW_s + bW_b \\
&= N\mu + N\left[\frac{b+N}{\sqrt{(b+1)(b+N)}}\right]\sigma \quad , \quad b \leq N\sigma \\
&= N\left(\mu + \frac{N}{4(b+1)}\,\sigma^2\right) + b \qquad , \quad b > N\sigma
\end{aligned}$$

Regime 3: $W_s > \mu + (b+2)W_b$

In this region, $f'(n) > 0$ for all n, and $f(n)$ is monotonically increasing with n. Therefore $F_{min} = f(1)$. The minimum distance condition that needs to be satisfied is

$$\begin{aligned}
f(1) = W_s + bW_b - \mu &> \sqrt{N}\sigma \\
W_s + bW_b &> \mu + \sqrt{N}\sigma
\end{aligned} \tag{5.21}$$

We wish to minimize $W_{tot} = NW_s + bW_b$ subject to this constraint. If we look at the above two expressions, it should be clear that this can be achieved by minimizing W_s subject to $W_s > \mu + (b+2)W_b$ and $W_b \geq 1$.

Consider choosing the minimum value of W_s for this regime, namely $W_s = \mu + (b+2)W_b$:

$$\begin{aligned} f(1) &= W_s + bW_b - \mu \\ &= \mu + (b+2)W_b + bW_b - \mu \\ &= 2W_b(b+1) \end{aligned}$$

To meet the minimum distance criterion, we need $f(1) \geq \sqrt{N}\sigma$. The minimum provisioning occurs at equality, so

$$\begin{aligned} 2W_b(b+1) &= \sqrt{N}\sigma \\ W_b &= \frac{\sqrt{N}}{2(b+1)}\sigma \end{aligned}$$

As long as $\sqrt{N}\sigma \geq 2(b+1)$, then $W_b \geq 1$ above and we can approximate the static and dynamic provisioning well by

$$\begin{aligned} W_s &= \mu + (b+2)W_b \\ &= \mu + \frac{b+2}{b+1}\frac{\sqrt{N}}{2}\sigma \end{aligned}$$

Then the total number of wavelengths required is

$$\begin{aligned} W_{tot} &= NW_s + bW_b \\ &= N\left(\mu + \frac{b+2}{b+1}\frac{\sqrt{N}}{2}\sigma\right) + \frac{b}{b+1}\frac{\sqrt{N}}{2}\sigma \end{aligned}$$

However, if $\sqrt{N}\sigma < 2(b+1)$, then the above expression gives a value of $W_b < 1$ and due to the integrality constraints we must choose either $W_b = 1$ or $W_b = 0$. Then from (5.21), the corresponding values of s would be

$$\begin{aligned} W_s &= \mu + \sqrt{N}\sigma - b \ , \ W_b = 1 \\ &= \mu + \sqrt{N}\sigma \qquad \ , \ W_b = 0 \end{aligned}$$

Are these valid choices of W_s for this regime? Consider first $W_b = 1$ and $W_s = \mu + \sqrt{N}\sigma - b$. Our assumption that $\sqrt{N}\sigma < 2(b+1)$ implies that $b > \frac{\sqrt{N}}{2}\sigma - 1$. We observe that

$$\begin{aligned} W_s &= \mu + \sqrt{N}\sigma - b \\ &< \mu + \sqrt{N}\sigma - \left(\frac{\sqrt{N}}{2}\sigma - 1\right) \\ &= \mu + \frac{\sqrt{N}}{2}\sigma + 1 \end{aligned}$$

while in this regime we require

$$\begin{aligned} W_s &\geq \mu + (b+2)W_b \\ &= \mu + b + 2 \\ &> \mu + \left(\frac{\sqrt{N}}{2}\sigma - 1\right) + 2 \\ &= \mu + \frac{\sqrt{N}}{2}\sigma + 1 \end{aligned}$$

which is contradictory. Therefore this is an invalid choice of W_s and W_b.

A similar analysis shows that $W_b = 0$ and $W_s = \mu + \sqrt{N}\sigma$ is a valid choice for this regime. Therefore we choose these values of W_s and W_b for when $b > \frac{\sqrt{N}}{2}\sigma - 1$. In this case, the total number of wavelengths is

$$W_{tot} = NW_s = N\left(\mu + \sqrt{N}\sigma\right)$$

Chapter 6

Conclusions

In this thesis, we have presented an investigation into the design of network architectures that trade off efficient resource allocation with nodal complexity. We have seen that a variety of differing approaches can be successful in greatly reducing node complexity at a minimal cost in terms of added resource allocation required.

We considered three angles of attack in reducing nodal complexity:

- Wavelength accessibility
- Switching granularity
- Hybrid static-dynamic provisioning

We first considered the base case of full reconfiguration. We derived novel results showing the necessary and sufficient minimum number of wavelengths required to support rearrangeably non-blocking traffic on a ring topology with wavelength conversion. We showed that full conversion at each node was not necessary to achieve this lower bound, and that a total of roughly 2 converters per wavelength was sufficient. We also showed that the converters could be arbitrarily located at any nodes within the ring using our converter shifting lemmas.

In the next chapter, we considered reducing node complexity by limiting wavelength accessibility and having some wavelengths optically bypass the majority of nodes. This reduced equipment costs by limiting the number of wavelengths dropped at each node and partitioned the wavelengths into two classes: "bypass" wavelengths resembling highways that were only accessible from certain hub nodes, and "local" wavelengths accessible at all nodes. We showed that using this approach, most nodes would only need to access a small number of local wavelengths, thus limiting the access costs at each node. The results of this chapter were also extended to torus and tree networks.

In the fourth chapter, we considered varying switching granularity as a method for reducing switching and multiplexing complexity. We observed that an all-optical switch need not operate by switching each wavelength individually; instead, multiple wavelengths can be switched together in a single waveband using the same switch port. We consider the number of wavelengths and wavebands (switch ports) required to support rearrangeably non-blockinng traffic on a star topology, and characterize the achievable region and the optimal tradeoff between wavelengths and wavebands. We showed that small increases over the minimum number of wavelengths required, combined with waveband switching, could yield a significant reduction in the number of switch ports required. We also show how to extend the approach to more general network topologies.

In our fifth chapter, we observed that even dynamic network traffic often exhibits a degree of predictability, and show that benefits can be derived from a partial static wavelength provisioning in conjunction with dynamic wavelength provisioning. We consider an asymptotic analysis for a large number of users and derive conditions for the network to be asymptotically non-blocking. We show that a shared link with a static provisioning of at least the mean performs as well as a fully dynamic system. We also show that by allowing waveband switching, a significant reduction in the total number of wavelengths required can be achieved (compared with individual wavelength-granularity switching) using only a small number of switches per node.

In future work, it would be interesting to investigate the extension of the results of Chapter 5 to include the effects of wavelength continuity and to investigate the role of wavelength converters. Some preliminary investigation has also shown that the results of Chapter 5 play a role in provisioning for blocking networks, where we design the network to achieve a target overflow probability. The question of whether target overflow probabilities can be linked to a call blocking probability (which distinguishes a single blocked call from multiple calls, for example) would be an interesting avenue to pursue.

Bibliography

[1] D. Banerjee and B. Mukherjee. Wavelength-routed optical networks: linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Trans. Networking*, 8:598–607, October 2000.

[2] R. A. Barry and P. A. Humblet. Models of blocking probability in all-optical networks with and without wavelength changers. *IEEE J. Select. Areas Commun.*, 14:858–867, June 1996.

[3] D. Bertsekas and R. Gallager. *Data Networks, 2nd Ed.* Prentice-Hall, New Jersey, 1992.

[4] A. Birman. Computing approximate blocking probabilities for a class of all-optical networks. *IEEE J. Select. Areas Commun.*, 14(5):852–857, June 1996.

[5] X. Cao, V. Anand, Y. Xiong, and C. Qiao. A study of waveband switching with multilayer multigranular optical cross-connects. *IEEE J. Select. Areas Commun.*, 21:1081–1095, September 2003.

[6] L.-W. Chen and E. Modiano. Efficient routing and wavelength assignment for reconfigurable WDM networks with wavelength converters. In *Proc. IEEE INFOCOM*, April 2003.

[7] L.-W. Chen and E. Modiano. Efficient routing and wavelength assignment for reconfigurable WDM networks. *IEEE/ACM Trans. Networking*, 13:173–186, February 2005.

[8] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: an approach to high bandwidth optical WANs. *IEEE Trans. Commun.*, 40:1171–1182, July 1992.

[9] R. B. Cooper. *Introduction to Queueing Theory, 2nd Ed.* North Holland, New York, 1981.

[10] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley-Interscience, 1991.

[11] H. M. de Oliveira and G. Battail. A capacity theorem for lattice codes on gaussian channels. In *Proc. ITS*, Sept 1990.

[12] A. F. Elrafaie. Multiwavelength survivable ring network architechtures. In *Proc. ICC '93*, pages 1245–1251, May 1993.

[13] A. A. Fredericks. Congestion in blocking systems - a simple approximation technique. *Bell Syst. Tech. J.*, 59:805–827, July-August 1980.

[14] O. Gerstel, G. Sasaki, S. Kutten, and R. Ramaswami. Worst-case analysis of dyanmic wavelength allocation in optical networks. *IEEE/ACM Trans. Networking*, 7:833–845, December 1999.

[15] R. Guerin and L. Y.-C. Lien. Overflow analysis for finite waiting-room systems. *IEEE Trans. Commun.*, 38:1569–1577, September 1990.

[16] R. Izmailov, S. Ganguly, V. Kleptsyn, and A. C. Varsou. Non-uniform waveband hierarchy in hybrid optical networks. In *Proc. IEEE INFOCOM*, April 2003.

[17] E. Karasan and E. Ayanoglu. Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks. *IEEE/ACM Trans. Networking*, 6:186–196, April 1998.

[18] M. Kovacevic and A. Acampora. Benefits of wavelength translation in all-optical clear-channel networks. *IEEE J. Select. Areas Commun.*, 14:868–880, June 1996.

[19] R. M. Krishnaswamy and K. N. Sivarajan. Design of logical topologies: a linear formulation for wavelength-routed optical networks with no wavelength changers. *IEEE/ACM Trans. Networking*, 9:186–198, April 2001.

[20] M. Lee, J. Yu, Y. Kim, C. Kang, and J. Park. Design of hierarchical crossconnect WDM networks employing a two-stage multiplexing scheme of waveband and wavelength. *IEEE J. Select. Areas Commun.*, 20:166–171, January 2002.

[21] L. Li and A. K. Somani. Dynamic wavelength routing using congestion and neighborhood information. *IEEE/ACM Trans. Networking*, 7:779–786, October 1999.

[22] P. J. Lin. *Wide Area Optical Backbone Networks.* PhD thesis, MIT, February 1996.

[23] D. A. Garbin M. J. Fischer and G. W. Swinsky. An enhanced extension to wilkinson's equivalent random technique with application to traffic engineering. *IEEE Trans. Commun.*, 32:1–4, January 1984.

[24] A. Narula-Tam, P. J. Lin, and E. Modiano. Efficient routing and wavelength assignment for reconfigurable WDM networks. *IEEE J. Select. Areas Commun.*, 20:75–88, January 2002.

[25] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 2002.

[26] R. Ramaswami and G. Sasaki. Multiwavelength optical networks with limited wavelength conversion. *IEEE/ACM Trans. Networking*, 6:744–754, December 1998.

[27] R. Ramaswami and K. N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Trans. Networking*, 3:489–500, October 1995.

[28] R. Ramaswami and K. N. Sivarajan. *Optical Networks: A Practical Perspective.* Morgan Kaufmann, 1998.

[29] S. Ramesh, G. N. Rouskas, and H. G. Perros. Computing blocking probabilities in multiclass wavelength-routing networks with multicast calls. *IEEE J. Select. Areas Commun.*, 20:89–96, January 2002.

[30] P. Saengudomlert. *Architectural Study of High-Speed Networks with Optical Bypassing.* PhD thesis, MIT, September 2002.

[31] P. Saengudomlert, E. Modiano, and R. G. Gallager. An on-line routing and wavelength assignment algorithm for dynamic traffic in a WDM bidirectional ring. In *Proc. JCIS*, March 2002.

[32] P. Saengudomlert, E. Modiano, and R. G. Gallager. Dynamic wavelength assignment for WDM all-optical tree networks. In *Proc. Allerton*, September 2003.

[33] P. Saengudomlert, E. Modiano, and R. G. Gallager. On-line routing and wavelength assignment for dynamic traffic in WDM ring and torus networks. In *Proc. IEEE INFOCOM*, April 2003.

[34] A. A. M. Saleh and J. M. Simmons. Architectural principles of optical regional and metropolitan access networks. *J. Lightwave Technol.*, 17:2431–2448, December 1999.

[35] J. M. Simmons, E. L. Goldstein, and A. A. M. Saleh. Quantifying the benefit of wavelength add-drop in WDM rings with distance-independent and dependent traffic. *J. Lightwave Technol.*, 17:48–57, January 1999.

[36] S. Subramaniam, M. Azizoglu, and A. Somani. A new analytical model for multifiber WDM networks. *IEEE J. Select. Areas Commun.*, 18:2138–2145, October 2000.

[37] M. N. S. Swamy and K. Thulasiraman. *Graphs, Networks, and Algorithms.* Wiley, New York, 1981.

[38] R. I. Wilkinson. Theories of toll traffic engineering in the U.S.A. *Bell Syst. Tech. J.*, 35:412–514, March 1956.

[39] Y. Zho, G. N. Rouskas, and H. G. Perros. A path decomposition algorithm for computing blocking probabilities in wavelength routing networks. *IEEE/ACM Trans. Networking*, 8:747–762, December 2000.