Wavelength Converter Placement in Survivable All-Optical Networks

by

Aaron So

A project report submitted to the
Department of Mathematics and Statistics
in conformity with the requirements for
the degree of Master of Science

Queen's University
Kingston, Ontario, Canada
September 2003

Abstract

This work studies the effect of wavelength converters at nodes of a survivable wavelength division multiplexed all-optical network. Simulations were performed to study the benefit of wavelength converters in survivable all-optical networks. Moreover, a wavelength converter placement algorithm for survivable networks, based on the utilization of wavelength converters, is proposed. The performance of this wavelength converter placement algorithm is tested against an existing converter placement algorithm which does not take network survivability into account.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Professor Glen Takahara for his patient guidance, encouragement and excellent advice throughout my first research endeavour. Also, I would like to express my sincere thanks to Professor Lorne Campbell for his financial support. Furthermore, it is my great pleasure to acknowledge Professor Fady Alajaji, Professor Peter McLane, Professor Chi-Hsiang Yeh, and again Professor Glen Takahara, for their marvellous lectures on various topics in communications. Their expertise and teaching enthusiasm have opened a door for me to the modern world of telecommunications. Finally, I would like to express my heartfelt thanks to all the faculty and staff at Queen's University for all the quality education and help in the past seven wonderful years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Despite the recent downturn in the communication industry, many researchers and industry leaders believe that the demand for bandwidth in optical networks will increase in the future, and that a higher bandwidth communications infrastructure has to be developed. An All-Optical Network is the architecture most likely to provide this infrastructure. The large bandwidth can be achieved through Wavelength Division Multiplexing (WDM), which combines separate signals on different wavelengths of light and transmits them along a single fibre. In current fiber optics networks, an optical signal has to be converted back to an electrical signal when it arrives at a switch. This conversion creates a bottleneck effect and in turn slows the overall transfer. An All-Optical Network, which performs switching and routing operations only in the optical domain, can increase the network performance dramatically [1](pp.1-5).

In an All-Optical Network, the optical signal is not converted back to an electrical signal during its journey. Therefore, without special hardware, a session is restricted to use the same wavelength on all of the links of its path. This wavelength continuity constraint is a significant problem in the development of All-Optical Networks. To ease this problem, a piece of hardware that can convert an optical signal from one wavelength to another was developed. However, this wavelength converter is still complex and expensive; moreover, wavelength conversion will cause signal attenuation. Thus, we would like to avoid wavelength conversion as much as possible [2]. Much research has been done on effective routing and wavelength assignment which assumes the absence of wavelength converters, while relatively little energy has been spent on investigating the performance improvement of an All-Optical Network when wavelength converters are introduced.

Due to the fact that an All-Optical Network is designed to carry a huge amount of information, a link or node failure could causes disastrous service disruption. As a result, network survivability is an active research area. In survivable networks, a connection is protected by a redundant protection path. If a link or a node fails in this connection, the traffic will be switched to the protection path. Hence, during connection setup, two paths have to be setup for a single connection. This means more network resources are consumed for each connection. In turn, the probability that no suitable route and wavelength can be found (blocking) for this connection also increases. There is no doubt that wavelength converters can be used to alleviate this problem. How much performance improvement can be contributed by wavelength converters to a survivable network is an interesting problem that is worthwhile to investigate. On the other hand, because a wavelength converter is an expensive piece of hardware, it is not economically feasible to equip every node with a wavelength converter. Thus, in the network designing stage, the number of available wavelength converters will usually be given as a constraint. It has been shown that strategic placement of a few wavelength converters will result in satisfactory performance [3]. The optimal placement of a limited number of wavelength converters in a survivable network is also an important and interesting problem that is worthwhile to investigate.

1.2 Project Overview

There are two goals in this project. The first goal is to investigate the performance improvement of survivable networks when wavelength converters are introduced in every node. The second goal is to determine the placement of a limited number of wavelength converters in a survivable network that gives sub-optimal performance. In Chapter 2, some background about All-Optical Network, routing and wavelength assignment algorithms and converter placement algorithms will be introduced. In Chapter 3, the concept of survivable routing is introduced. Moreover, a newly developed survivable routing algorithm, called Short Leap Shared Protection (SLSP) Framework [4], is reviewed. Furthermore, a slightly enhanced version of SLSP that can take advantage of wavelength converters in the network is introduced. We are interested in investigating the performance improvement of a survivable network with full wavelength conversion capability. The measure of network performance is in terms of blocking probability and unfairness. In order to analyze the performance improvement, simulations were performed. In the simulations, Short Leap Shared Protection (SLSP) Framework is used. Two sets of simulations were performed. Both sets used the same network topology. In the first set of simulations, no wavelength converters were present. The number of available wavelengths and load were set as a variable, and the blocking probability and unfairness for each given number of wavelengths and load were recorded. The second set of simulations was basically the same except wavelength converters were available at every node. In the second set of simulations, the utilization of each wavelength converter was also recorded. The utilization of each wavelength converter is an important piece of information that can be used to determine the strategic placement of a limited number of wavelength converters. In Chapter 4, the sub-optimal placement of a limited number of wavelength converters is investigated. Sub-optimal converter placement algorithms have been developed in [2][5][6][7][8][9] for non-survivable networks. A new converter placement algorithm for a survivable network is developed and will be presented in this report. Its performance is compared with an existing converter placement algorithm, called Minimum Blocking Probability First (MBPF) [8], which does not take survivability of the network into account. The measure of the converter placement performance is blocking probability given the number of wavelength converters added. In order to analyze the performance, simulations were performed. In this project, three different converter placement algorithms are tested via simulations. For each simulation, the same network topology, number of wavelengths and load are used so that the results are comparable. In Chapter 5, the descriptions of the performed simulations and the results are presented and discussed. Chapter 6 gives the conclusions and outlines some future work in this area.

1.3 Contributions

The following is a list of the major contributions of this work.

- Performance analysis of the Short Leap Shared Protection (SLSP) Framework with and without wavelength converters were performed via computer simulations. The results are in terms of blocking probability and fairness with respect to the load and number of wavelength used.

- A slightly enhanced version of SLSP that can take advantage of wavelength converters in the network was developed.

- Heuristics for converter placement in a survivable network based on converter utility in each node were developed. Their performance was tested against an existing converter placement algorithm which does not take network survivability into account.

- Simulation programs written in Java were developed to study the blocking probability, fairness, and converter utilization of a network given the network topology, load, number of wavelengths and the location of wavelength converter(s).

- A computer program written in Java which implements the minimum blocking probability first (MBPF) wavelength converter placement algorithm was also developed.

Chapter 2

Literature Review

2.1 General All-Optical Networks

An All-Optical Network (AON) is basically a set of nodes connected by internodal network links. Each internodal network link is composed of a set of fibers (usually 2, one for each direction), and each fiber can be occupied by a set of λ-channels. A λ-channel is the smallest entity in the optical spectrum partition, and each λ-channel is assigned a distinct wavelength.

When a connection request between a source and destination node arrives, a path, which comprises of a sequence of nodes and links, and a suitable λ-channel have to be found for this connection. Routing involves determining a suitable path for this connection, while wavelength assignment involves allocating an available wavelength for this connection. Due to the fact that joint routing and wavelength assignment is computationally heavy, they are usually treated separately.

After a routing algorithm determines a suitable path for a connection, the wavelength cannot be assigned arbitrarily. The wavelength assignment of a connection is subject to the wavelength continuity constraint and the distinct channel assignment constraint. Due to the fact that a connection in an All-Optical Networks stays in the optical domain for its entire journey, the wavelength of an optical connection remains the same on all links its traverses from source to destination. This is called the wavelength continuity constraint. Moreover, for all connections that share a common fiber, they must be assigned distinct wavelengths; otherwise, the signals will be mixed. This is called the distinct channel assignment constraint.

In an All-Optical Network, the number of available wavelengths is limited. Since all optical connections cannot violate the wavelength continuity constraint and distinct channel assignment constraint, this makes routing and wavelength assignment a fundamental control problem in All-Optical Networks [1] (pp.114-118).

2.2 Routing and Wavelength Assignment(RWA)

With no wavelength converter in the network, the routing and wavelength assignment algorithms are classified based on the network traffic type: static or dynamic. For static traffic, the traffic requirements are given as a prescribed set of point to point connections. These connections are assumed to remain in place forever (a relatively long period of time). In contrast, for dynamic traffic, a sequence of connection requests arrives in some random fashion, and a successful connection will be released after a random amount of time. In both traffic types, as described before, the joint routing and wavelength assignment problem is computationally heavy, it is usually necessary to solve them separately and produce a sub-optimal result. This trade off between complexity and optimality is worthwhile because it is much easier to accomplish, and the results are generally satisfactory for engineering purposes.

2.2.1 RWA in Static Traffic

In [10], a theoretical approach is used to solve the RWA problem for static traffic. The problem is cast as a classic graph coloring problem. Given a network topology and a prescribed set of point to point connections, the optimal solution of the RWA problem consists of finding a feasible solution, in which all connections satisfy the wavelength continuity constraint and distinct channel assignment condition, that uses the minimum number of wavelengths.

To find the optimal solution, a path interference graph, G, is needed. To construct G, a set of possible paths for each prescribed connection is computed. The set of possible paths for each connection can be controlled by the network designer. It can be the k-shortest paths, or a set of path selected by other network criteria. Each selected path represents a vertex in the path interference graph, G. Two vertices of G are joined by an edge if the paths that they represent share a common fiber.

Having constructed G, finding the optimal solution of the RWA problem consists of selecting one vertex (a path) for each prescribed connection, forming a subgraph of G, and choosing a color for each selected vertex (representing a wavelength for the connection) so that adjacent vertices are assigned different colors. If the assignments are made with a minimum number of colors (wavelengths), this is known as minimal vertex coloring of the subgraph of G. It is a classic (and difficult) problem in graph theory, which is computationally heavy. However, it gives us the optimal solution.

To reduce the complexity of the problem, we have to do routing (subgraph construction) and wavelength assignment (vertex coloring) separately. First, we can choose a path (vertex) for each connection according to some routing criterion (e.g. shortest path), and form the subgraph of G, induced by this choice. Then we can do the minimal vertex coloring (wavelength assignment) of the chosen subgraph. This reduces the complexity of the problem tremendously, but it only gives us a sub-optimal solution. However, it is generally satisfactory for engineering purposes.

2.2.2 RWA in Dynamic Traffic

Under dynamic traffic, connection requests arrive to the network in a random fashion. Depending on the network state, an arrived connection request may or may not be accommodated. If a connection request cannot be accepted, it is blocked. The network state depends on the set of all connections that are currently in progress. Thus, the network state evolves randomly in time as new connection requests are admitted and active connections are released. Due to the dynamic traffic nature, in performing routing and wavelength assignment, an algorithm must be executed in real time. Because of the real-time nature of the problem, algorithms for RWA in a dynamic traffic environment must be very simple. As described before, one way to simplify the problem is to do routing and wavelength assignment separately.

In [8], a popular routing algorithm called fixed alternate routing (FAR) is summarized. Using FAR, each source and destination pair is assigned a set of edge-disjoint paths which are calculated before the network is in operation. For each set of predefined paths, the paths are numbered according to some criterion (usually based on number of hops). When the network is in operation and a connection request between a source and destination pair arrives, the predefined paths are tried one by one based on their number. If the connection can be established by using one of the predefined routes, the one with the least number will be chosen. Otherwise, the connection is blocked. FAR is simple, fast, and its performance is usually satisfactory. However, in some cases, even though there exists a path between the source and destination pair, FAR cannot establish the connection successfully because the desired path does not belong to its predefined path list.

In [1](pp.429), another popular routing algorithm called shortest-path with deletions is summarized. By using this routing algorithm, when a connection request between a source and destination pair arrives, each wavelength is examined. For each wavelength, the topology is modified. If we are examining wavelength 1, all links in the network with wavelength 1 occupied are deleted. Then Dijkstra's shortest path algorithm [11] (pp.401-403) is performed on the modified network to find a path between the source and destination pair. After all wavelengths are examined, the path with the least cost is selected. If no path can be found for all wavelengths, the connection is blocked. This fully adaptive routing algorithm can outperform FAR. However, the connection delay is much higher, and it requires the network to exchange much more control messages. The routing algorithm only selects the path for a connection. The wavelength to be used by this connection has to be determined. The most simple wavelength assignment algorithm is called Random Fit (RF) [12]. If more than one wavelength is available for a given path, it just selects one randomly. Another simple but much more effective wavelength assignment algorithm is called First Fit (FF) [13]. This algorithm numbers all the available wavelengths before the network is in operation. If more than one wavelength is available for a given route, the First Fit algorithm will select the one with the least number. The idea behind the First Fit algorithm is to avoid network resources fragmentation. It encourages connections to use wavelengths with a low number so that the high number wavelengths can be used if the low number wavelengths are fully occupied. Based on the same rationale, another wavelength assignment algorithm called Most-Used (MU) [14] was proposed. By using Most-Used, if more than one wavelength is available for a given route, the most utilized wavelength will be selected. It avoids network resources fragmentation by encouraging a connection to use wavelengths that are already in use elsewhere in the network so that network resources for other wavelengths can be reserved.

2.3 Survivability - Protection and Restoration

Due to the vast amount of bandwidth being carried in an optical network, a single link or node failure can cause the loss of a huge amount of information. Thus, protection and restoration are important issues in optical networks. For a network to be survivable, its topology must allow rerouting around a fault condition. Moreover, redundancy is vital to ensure that traffic is not disrupted in the case of fiber or node failure. Basically, there are two types of protection: link protection and path protection.

As the name implies, link protection protects any network connection against link failure. In [15], a scheme called (1+1) Optical Protection is proposed. By using this scheme, each link in the network is composed of two fibers; one is the working fiber, while the other is the protection fiber. Both fibers carry the same signal, and the destination node receives the signal from the working fiber in normal circumstances. If the working fiber is cut, the destination node will switch and receive from the protection fiber. In this way, any single link failure can be restored. Other link protection schemes such as (1:1) Optical Protection and (1:N) Optical Protection [1] (pp.630-631) are similar to (1+1) Optical Protection scheme but they involve more signaling between the nodes around the failing area.

Besides link protection, path protection is a protection mechanism that protects the entire path from failing. When a connection request between a source and destination arrives, a path, which comprises of a sequence of nodes and links, is set up for this connection. Under normal circumstances, traffic is carried by this path. We call it the working path.

For protection purpose, another path between the same source and destination is also setup. If the working path fails, the traffic will be switched to this path to restore the traffic. We call this path the protection path. A protection path can be setup as dedicated or shared. For dedicated protection, a protection path is set up for each working path. Thus, when a working path fails, the traffic switches to its dedicated protection path. With shared protection, a protection path is shared by a number of working paths. If one of the working paths fails, the traffic can be switched to this shared protection path. If two or more of these working paths fail, only the highest priority working path will be restored. In Chapter 3, a shared protection survivable routing algorithm, called iterative two step approach (ITSA) [16] will be reviewed. In addition, a more advanced survivable routing algorithm, that uses the concept of

ITSA, called Short Leap Shared Protection (SLSP) Framework [4] will be described in detail.

In [17], another survivable routing approach which uses a linear integer programming optimization technique is proposed. Even though [17] proposes an algorithm that reduces the computation time of the optimization, it is still not quite feasible for a real time system in dynamic traffic. In [18], an algorithm is proposed that finds a survivable route by finding a logical ring in the network. However, depending on the network topology, such a strategy is not always possible. Thus, for some topologies, it is not a desirable survivable routing algorithm.

2.4 Wavelength Converter Placement

To alleviate the burden of the wavelength continuity constraint in wavelength assignment, optical wavelength converter have been developed. A wavelength converter is a device that converts an incoming wavelength to a different outgoing wavelength all in the optical domain. A network node equipped with a number of wavelength converters is called a wavelength-convertible switching node (note: a wavelength converter can be occupied by only one connection at a time).

2.4.1 Hardware Structure

There are two popular structures to configure converters in a network node: share-per-node and share-per-link. In the share-per-node structure[5], all the converters at the switching node are collected in the converter bank (CB) which can be accessed by all the incoming connections of that switching node.

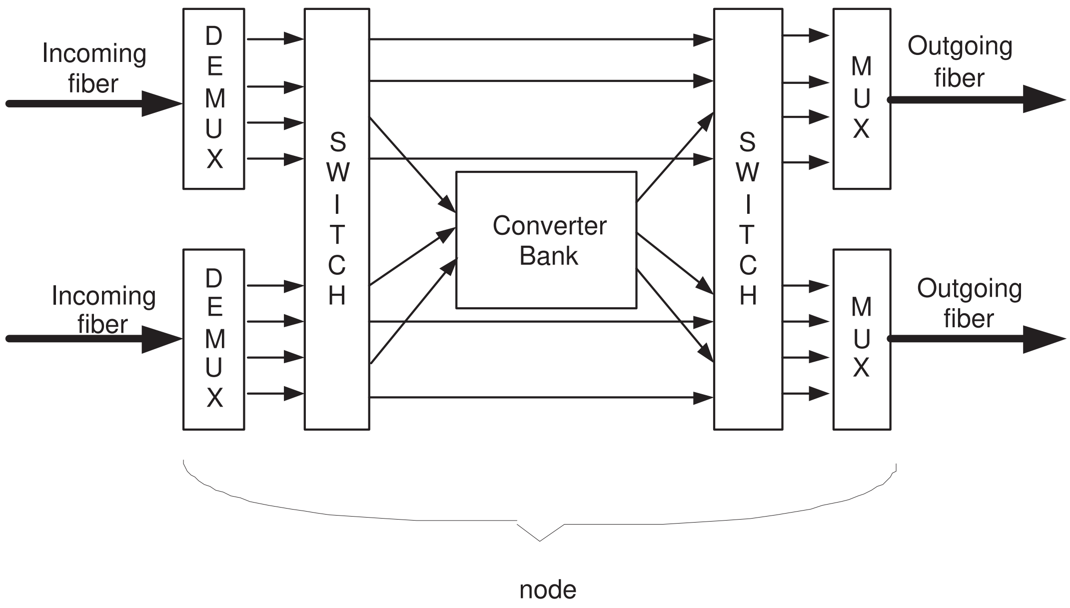

Figure 2.1: Share-per-node structure: all available wavelength converters are shared in the converter bank.

In a share-per-link structure [19], each outgoing fiber owns a dedicated converter bank which can only be accessed by the connections of a particular pair of incoming and outgoing fibers (See Figure 2.2).

In terms of hardware, the complexity of the share-per-link structure is much simpler than that for the share-per-node structure. However, for the share-per-link structure, the wavelength converters may not be fully utilized in dynamic traffic conditions because they are dedicated to a fixed incoming fiber. If a connection in the other fiber needs wavelength conversion and its CB is fully occupied, it cannot use the other CB since it is dedicated for the other fiber. In Chapters 3,4,5, and 6 of this report, we do not care which hardware structure is used. We are only interested in whether a node has full wavelength conversion capability (i.e., if a node is equipped with a wavelength converter, we mean this node has enough wavelength converters

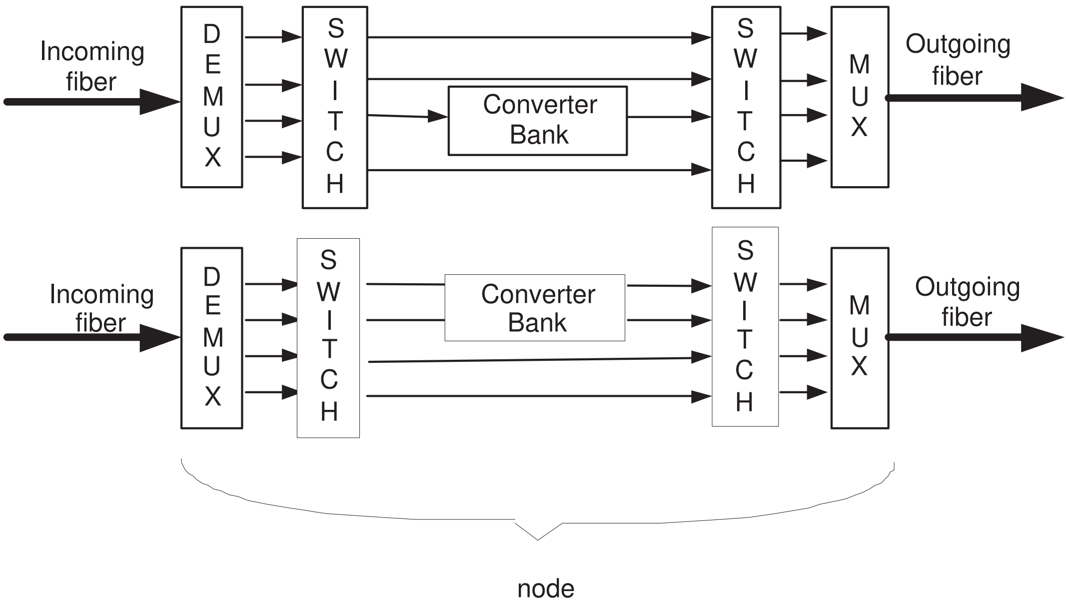

Figure 2.2: Share-per-link structure: wavelength converters are dedicated to a particular incoming and outgoing fiber pair.

in its converter bank such that all incoming signals have the ability to change their wavelength simultaneously). However, for the remainder of this chapter the hardware structure of a conversion capable node is specified.

2.4.2 Wavelength Converter Placement Under Static Traffic

In static traffic, as defined in Section 2.2.1, given a prescribed set of connections, a route and a suitable wavelength can always be found for all connections if the number of wavelengths available is not bounded. If the number of available wavelengths is fixed, and no one RWA algorithm can satisfy all connections without using more wavelengths, wavelength converters may help solve the problem. In [20], a routing, a wavelength assignment, and a wavelength converter placement algorithm are proposed. In this scheme, the number of available wavelengths is given (the number of available wavelengths must be greater than the number of channels in the most heavily loaded link). The objective of this scheme is to find a set of routes, allocate wavelength converters in the network, and assign wavelength(s) to each of the routes such that all the connections in the prescribed set are satisfied, and the number of wavelengths used does not exceed the given number of available wavelengths. Since the traffic is static, the algorithms only have to determine the number of converters to be placed in the converter bank of each node. Thus, the share-per-link configuration is preferred because of its simplicity.

The routing algorithm tries to balance the load in each link in the network so that no one particular link is very congested. This is important because the load of the most congested link determines the minimum number of required wavelengths. That is, the number of available wavelengths has to be equal to or greater than the number of connection in the most congested link. After routing, a simple wavelength assignment algorithm, which assumes the number of available wavelengths is unbounded, is executed. After this routing and wavelength assignment, if the number of required wavelengths is less than or equal to the number of available wavelengths, the RWA problem is solved, and no wavelength converter is needed. Otherwise, a wavelength converter placement algorithm is needed to reduce the number of required wavelengths to the number of available wavelengths. If the number of required wavelengths is WR, and the number of available wavelengths is W, we have to reconfigure the connections that uses wavelength number $W+1$ to WR. For each of these connections, the connection will try using wavelength 1 to W one by one, and count the number of conflict links that are introduced by this reassignment. The wavelength that introduces the least conflict will be used. Then wavelength converters will be placed in the network to resolve the conflict introduced by the reassignment. In this way, all the prescribed connections will be satisfied, and the number of wavelengths required will not exceed the number of available wavelengths. However, this algorithm assumes that the supply of wavelength converter is unbounded, which may be unrealistic.

In [19] , the problem of having a limited number of wavelength converters is addressed. Basically, a converter placement algorithm which assumes no upper bound on the number of available converters is used. After the converter allocation, if the number of converters used is higher than the number of available converters, some converters have to be removed, and some connections in the prescribed set have to be sacrificed. In [19], this situation is cast as an optimization problem with the objective of maximizing the traffic going through the entire network.

The methods in [19] and [20] can produce result in a reasonable amount of time, and the performances are generally satisfactory. However, they do not guarantee an optimal solution (an optimal solution is a solution that use the minimum number of wavelength converters that can produce the same result as if there were wavelength converters in every node). In [22], it is shown that finding an optimal solution in a network with mesh topology is NP-complete, and [23] shows that this problem is equivalent to the famous minimum vertex cover problem [24], which is also NP-complete.

In [22], an interesting situation is pointed out. If we put a wavelength converter (full range) in a single node in a WDM bidirectional ring, the result is equivalent to having wavelength converters in all nodes. In order words, an optimal result can be obtained by simply putting a converter in only one node. Inspired by this discovery, researchers were interested in searching for polynomial time algorithms that can find an optimal solution for some special topologies. In [21], two polynomial time algorithms are presented which can find the optimal solutions for two topologies: tree and tree of rings. The algorithm reduces the converter placement problem for these special topologies to a minimum vertex cover problem, and then reduces it to the minimum sufficient set problem [24]. Finally, it solves the special case of the minimum sufficient set problem in polynomial time. This discovery is significant because tree and tree of rings topologies are widely used in the telecommunication industry. Such algorithms are very useful in designing and minimizing resource use in an All-Optical Network with tree or tree of rings topologies.

2.4.3 Wavelength Converter Placement Under Dynamic Traffic

Under dynamic traffic, as described in Section 2.2.2, connection requests arrive to the network in a random fashion. Depending on the network state, an arrived connection request may or may not be accommodated. If a connection request cannot be accepted, it is blocked. The performance of an RWA algorithm in dynamic traffic condition is often measured in terms of blocking probability. By adding wavelength converters and using the same RWA algorithm, the blocking probability should decrease, at least not increase. However, given a limited number of wavelength converters, placing them in different locations will give different results. Thus, the performance of a wavelength converter placement algorithm is measure in terms of blocking probability given the number of wavelength converter added.

In [6], an optimal converter placement algorithm is presented for linear network.

Given a dynamic traffic requirement, an optimal placement of a limited number of wavelength converters minimizes the overall blocking probability of the network. The study in [6] shows that uniform spacing of converters in an linear network is optimal when link loads are uniform and independent. Moreover, when the link loads are nonuniform, the optimal placement of the converters can be found by a dynamic programming algorithm. The works of [6] can only apply to linear networks. For network with mesh topology, optimal placement of wavelength converters is computationally unfeasible. Thus, sub-optimal placements are often proposed for networks with mesh topologies.

In general, there are two sub-optimal strategies in placing wavelength converters in the network. One uses simulations to generate network statistics and determine the placement, while the other uses a mathematical model to calculate the approximate blocking probability of a network, and calculates the placement of the wavelength converters that minimizes the overall blocking probability.

In [2] and [5], statistics generated from simulations to find a placement for a given number of wavelength converters are used. In addition to the generated statistics, [5] use optimization techniques to find a placement for a limited number of wavelength converters. In particular, [5] uses the share-per-node structure and calculates the number of wavelength converters in the converter bank in every node that maximize the utility of each wavelength converter. In the simulation, each node is assumed to have full wavelength conversion capability. For each node, the percentage of time that x wavelength converters in the converter bank are used simultaneously is recorded. After the simulation is completed, the utilities of wavelength converters are used in the optimization calculation. The objective of the optimization is to maximize the overall utility of all converters, and the constraint is the total number of available wavelength converters.

In [7] [8] and [9], another approach is used to find the placement of wavelength converters. Mathematical models are developed in calculating the blocking probability of the network given some wavelength converters placed in some locations. In these papers, a wavelength converter means a node with full range conversion capability. In [7], an exhaustive search method is used to search for a placement that minimizes the blocking probability. However, it may not be always feasible especially in a large network. In [8] and [9], a more feasible method is presented. An approximate mathematical model that calculates the blocking probability of the network given a limited number of wavelength converters placed in some locations is developed. The algorithm places the wavelength converters one by one. All nodes that are not equipped with a wavelength converter are called candidate nodes. The first wavelength converter is placed in the candidate node if placing a wavelength converter in this node will result in the lowest blocking probability when compared with placing a wavelength converter in another candidate node. After the first converter is placed, another converter is placed in another candidate node based on the same strategy. This process repeats until all the available wavelength converters are used. This algorithm will be explained in more detail in Chapter 4.

Chapter 3

RWA in Survivable Networks

In this chapter, the notion of shared protection is introduced. Then, a shared protection survivable routing algorithm called Iterative Two Step Approach (ITSA) [16] will be discussed. In addition, a more advanced survivable routing algorithm, that uses the concept of ITSA, called Short Leap Shared Protection (SLSP) Framework [4] will be described in detail. Moreover, the First Fit (FF) wavelength assignment algorithm will be reviewed. In Sections 3.2, 3.3 and 3.4, the network is assumed to have no wavelength converters. In Section 3.5, a slightly enhanced version of SLSP-FF RWA algorithm which can take advantage of wavelength converters in the network will be introduced.

3.1 Protection and Restoration

Due to the vast amount of bandwidth being carried in an optical network, a single link or node failure can cause the loss of a huge amount of information [1] (pp.607-608). Thus, protection and restoration are important issues in optical networks. For a network to be survivable, its topology must allow rerouting around a fault condition. Moreover, redundancy is vital to ensure that traffic is not disrupted in the case of fiber or node failure. Path protection is a mechanism that protects the entire path from failing. For each connection, a working path and a redundant protection path are set up. Under normal circumstances, traffic is carried in the working path. If the working path fails, the traffic switches to the protection path. Since the protection path will be utilized only if its working path fails, some network resources can be saved if protection paths can share their resources.

With shared protection, protection resources are shared by a number of protection paths. If one working path fails, the traffic can be switched to the protection path without any problem. If two or more connections which use the same protection resources have their working paths failed, only the highest priority working path will be restored. Thus, for 100% restorability after a single link failure, only a subset of all connections can share a particular link for their protection paths. This is described as the Shared Risk Link Group Constraint.

3.1.1 Shared Risk Link Group (SRLG) Constraint

For 100% restorability after a single link failure, if two working paths share a common link, their protection paths cannot share any resources. For example, consider Figure

3.1.

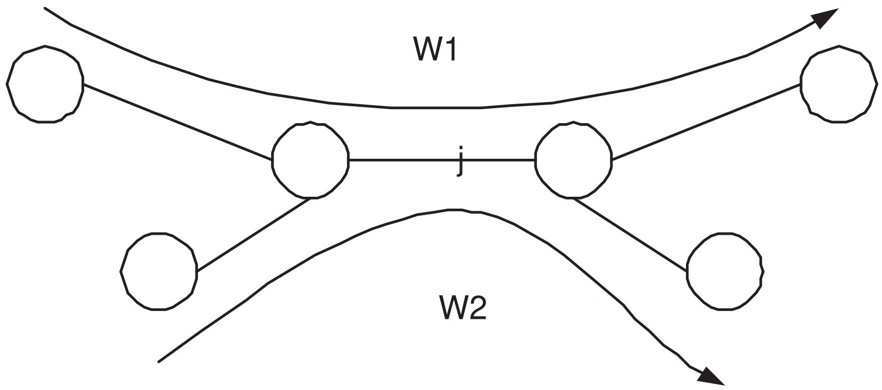

Figure 3.1: SRLG example.

Since $W1$ and $W2$ use a common link j, the protection paths of $W1$ and $W2$ should not share any network resources. If link j fails, both $W1$ and $W2$ will fail. The traffic of $W1$ and $W2$ will be switched to their protection paths. If the protection paths of $W1$ and $W2$ share some resources, only one of $W1$ or $W2$ will be restored. Thus, 100% restorability cannot be achieved. In other words, if 100% restorability after a single link failure is required, the protection path of $W1$ and $W2$ cannot share any network resources.

3.2 Survivable Routing - Iterative Two Step Approach(ITSA)

Survivable routing is defined as the adaptive path selection algorithm that derives a working and a protection path for any source and destination under dynamic network traffic. In this section, a shared protection survivable routing algorithm, called Iterative Two Step Approach (ITSA) [16] is reviewed. For a given source and destination pair, the algorithm inspects the k-shortest paths between this source and destination pair one by one. The k-shortest path is a set of paths for a source and destination pair which includes the shortest path, the second shortest path, the third shortest path,..., and the k^{th} shortest path. Each path in the k-shortest path set is treated as a working path. Given it is the working path, the best protection path for this working path can be found. In [16], the working path and protection path are assumed to use the same wavelength which is not necessary in practice. However, in this report, we use this assumption in our description. For all feasible working path and protection path pairs found, the pair which has the minimum cost is selected. Moreover, this algorithm also maximizes the resource sharing for protection paths while maintaining 100% restorability after a single link failure.

3.2.1 Network Model

Each link in the network is comprised of m fibers for each direction. Hence, up to m connections (all in the same direction) using the same wavelength can use this link. Moreover, if there are n wavelengths, we can treat the network as n identical single wavelength networks. Blocking occurs when no path is available for all n networks. For each copy of the network, each link can be occupied by m connections per direction. Since we assume that both working path and protection path use the same wavelength, let us assume that only one wavelength is employed in the network. Moreover, let us assume each connection occupies a single wavelength channel. For example, one direction of link j can be modeled as in Figure 3.2.

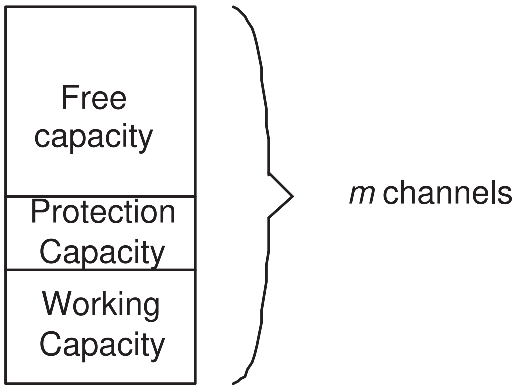

Figure 3.2: Capacity Allocation of Link j.

In Figure 3.2, the "protection capacity" is the number of channels currently allocated to protection paths while the "working capacity" is the number of channels currently allocated to working paths.

3.2.2 Two Step Approach for Shared Protection

Step 1 : Find a working path

When a connection request arrives, a working path between the source and destination has to be found. The weight of each link j in the network is defined as follows:

$$cw_j = \begin{cases} \infty & \text{if link } j \text{ has no more free capacity} \\ c_j & \text{otherwise} \end{cases} \tag{3.1}$$

where c_j is the cost of using link j (c_j can be either constant or depend on the dynamic link state).

After defining the weight of each link in the network, the k-shortest paths between a source and destination can be found by Yen's algorithm [25].

Step 2 : Given a working path, W, find a protection path

Given a working path, W, we can find the amount of protection capacity in each link that is prohibited from sharing with any protection path of W. (i.e, the amount of protection capacity that is subject to the SLRG constraint in each link). Let us define the following terms:

LW : Set of links of the given working path,W.

WD_W^l: Set of working paths traversing link l, where $l \in LW$.

SB_l^j: number of working paths in WD_W^l that have their protection paths traverse j.

S_j^W: Amount of protection capacity on link j that is prohibited from sharing with any protection path of working path W.

For each link j, S_j^W can be calculated as $S_j^W = max_{l \in LW}(SB_l^j)$.

Once we have calculated S_j^W, we can then calculate the weight of each link for protection purposes. The weight of each link j is now defined as follows:

$$cp_j^W = \begin{cases} \infty & \text{if link } j \text{ has no more free capacity and all protection} \\ & \text{capacity in link } j \text{ is subject to the SLRG constraint} \\ & \text{OR } j \text{ belongs to } LW \\ c_j & \text{if link } j \text{ has free capacity and all protection} \\ & \text{capacity in link } j \text{ is subject to the SLRG constraint} \\ 0 & \text{if link } j \text{ has some protection capacity that} \\ & \text{is not subject to the SLRG constraint} \end{cases} \quad (3.2)$$

Given the working path, W, we can update the weight of each link j in the network with cp_j^W. By using Dijkstra's algorithm [11] (pp.401-403), we can find the best protection path for this working path.

3.2.3 Iterative Two Step Approach

The idea of ITSA is simple. When a connection request between a source and destination arrives, the weight of each link is updated by using Step 1 of the above Two Step Approach. Then, the k-shortest paths are calculated by using Yen's algorithm [25]. For each path in the k-shortest path set, the best protection path can be found by Step 2 of the above Two Step Approach. Then the total cost of the working path and protection path is recorded. After all k-shortest paths are inspected, the one with the least total cost will be selected.

3.3 Short Leap Shared Protection (SLSP) Framework

Short Leap Shared Protection (SLSP) is another survivable routing algorithm that uses the concept of the ITSA algorithm [4]. SLSP divides the working path of a connection into several overlapping segments, called P-domains. The two end points of a P-domain are treated as source and destination, and a working path and a protection path are found by using a slightly modified version of ITSA. This procedure repeats for all P-domains. The concatenation of the working paths for all P-domains becomes the final working path, and the working path of each P-domain is protected by its own protection path. Hence, if a link or node fails, only the P-domain that the failed link or node is resided will need to switch the traffic from the working path to the protection path. The rest of the connection remains unchanged. Unlike ITSA, which working path and protection path can use different wavelengths, SLSP requires the protection path of each domain uses the same wavelength as the working path. This will be explained in Subsection 3.3.2.

3.3.1 Allocation of P-domain

The P-domain allocation can be done in two phases.

Phase 1 : Off line preparation.

For each source and destination pair, pre-calculate the k-shortest paths based on the number of hops. This can be done by using Yen's algorithm [25]. For each such path, pre-plan some overlapping segments. For example, consider Figure 3.3.

Figure 3.3: Definition of P-domain.

PSL(x) = xth Path Switched Label Switch Router

PML(x) = xth Path Merged Label Switch Router

Each PSL(x)-PML(x) pair defines a potential P-domain, which might be chosen to make the connection according the network conditions. The overlapping PSL(x)-PML(x) pair segments have the same distance (number of hops), except possibly at the last P-domain. We call the distance between PSL(x) and PML(x) the diameter of this P-domain. The notation "SLSP-n" defines the type of SLSP which has domain diameter equals n hops.

When a link or node failure occurs in the working path of a P-domain, the Path Switched Label Switch Router (PSL) will switch the working traffic to the protection path of that P-domain, and the Path Merged Label Switch Router (PML) will merge the traffic from the protection path to the working path of the next P-domain.

In Figure 3.3, each solid line represents a link which will be used, if available, in the working path. For each dashed line, the connection between its two end points is found adaptively based on network condition. If the links represented by the solid lines are all available, and connections can be found between the two end points of each dashed line, a working path is successfully found.

Furthermore, in Figure 3.3, PSL(x)-PML(x), where $x = 1, 2, 3$, will perform the switching and merging operation when failure occurs. PML(0) and PSL(4) are only used to define the first and last links of the working path.

Phase 2 : Real time path searching.

When a connection request arrives, update the weight of each link in the network according to their capacity availability. Then, do the following for each pre-planned path.

1. Use a slightly modified version of the Two Step Approach to find a working path and protection path for each potential P-domain.

    - For each potential P-domain, if the link PSL(i) - PML(i-1) is available, proceed; otherwise try the next pre-planned path.

- For each potential P-domain, find a path between PML(i-1) and PSL(i+1) by an adaptive algorithm (e.g. Dijkstra's Algorithm). If a working path is found for each potential P-domain, proceed; otherwise try the next pre-planned path.

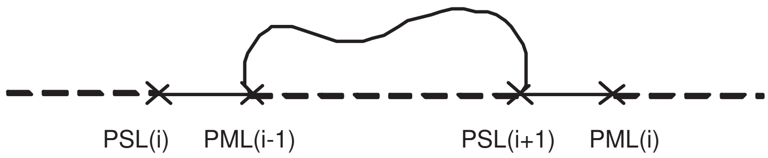

Figure 3.4: Finding a working path for each domain.

- For each potential P-domain, given a working path (now the working path is the concatenation of the links PSL(i)-PML(i-1) and the paths we just found in the previous step), update the weight of each link in the network based on the SLRG constraint. Then, find a protection path connecting PSL(i) and PML(i) for each i. If no protection path can be found for some i, then try the next pre-planned path.

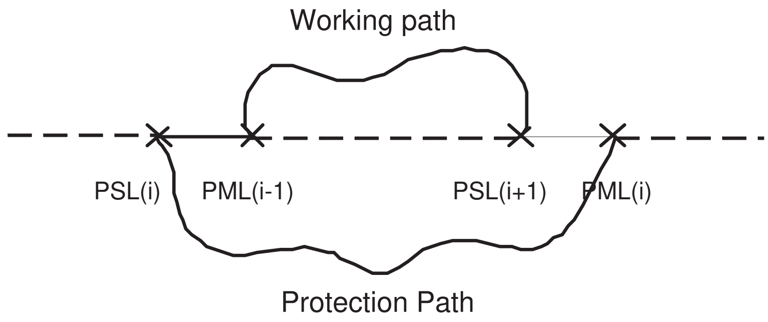

Figure 3.5: Finding a protection path for each domain.

2. Calculate and record the total cost of the working path and protection path for all potential P-domains for this pre-planned path.

If no pre-planned path can be used to allocate working and protection resources successfully, the connection request is blocked. Otherwise, select the pre-planned path with the least total cost.

3.3.2 100% Survivability After Single Link or Node Failure

After a working path and a protection path are selected for each P-domain, the selected working path can start carrying traffic. If any single link fails, 100% survivability can be achieved. Consider the example in Figure 3.6.

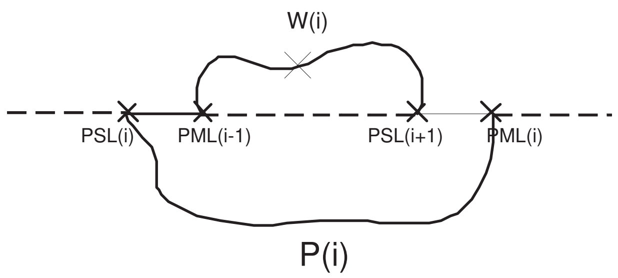

Figure 3.6: Restoration after a single link or node fails.

If any link in the working path segment W(i) fails, the traffic will switch to the protection path segment P(i) of this P-domain, while the rest of the traffic will remain unchanged. For protection against a single node failure, in most cases, the situation is identical to that of a single link failure. If any node, excluding PSL(i), in the working path segment W(i) fails, the traffic will switch to the protection path segment P(i) of this P-domain, while the rest of the traffic will remain unchanged. If PSL(i) fails, the working traffic of the previous P-domain will be switched to the protection path segment of that P-domain (see Figure 3.7). Like other cases, the rest of the traffic will remain unchanged.

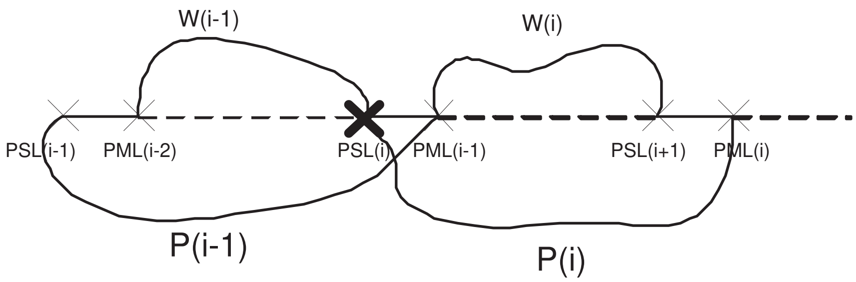

Figure 3.7: Restoration after PSL(i) fails.

Finally, there is one more remark for protection switching when there are more one wavelength are available. In Figure 3.7, if PSL(i) does not have a wavelength converter, P(i) and W(i-1) have to use the same wavelength at PSL(i). This is necessary because of wavelength continuity constraint. When a protection switch occurs, traffic from W(i-1) can be relayed to P(i) using the same wavelength. By the same token, if PML(i-1) does not have a wavelength converter, P(i-1) and W(i) have to use the same wavelength at PML(i-1). In the absence of wavelength converters in the network, this explains why the working path and the protection path of a connection have to use the same wavelength.

3.4 Wavelength Assignment

In the above sections, two survivable routing algorithms were discussed (ITSA and SLSP). In the description of the algorithms, we assumed only one wavelength is available in the network. Thus, wavelength assignment is not an issue. However, if more than one wavelength is available, a wavelength assignment protocol has to be employed. As described before, joint routing and wavelength assignment is computationally heavy; as a result, we treat routing and wavelength assignment separately.

Among all wavelength assignment schemes, the First Fit (FF) [13] scheme performs relatively well and is preferred in practice for its simplicity. The FF scheme numbers each available wavelength before the network is in operation. After the routing algorithm is run, a suitable wavelength which has the least number will be assigned for this connection. The idea of the FF scheme is to reuse the in-use wavelengths as much as possible and in turn avoid network resource fragmentation. In the absence of wavelength converters, if there are n wavelengths available, we can view the network as n identical single wavelength networks. Each network representing a wavelength is numbered. By running SLSP (or ITSA), when a connection request arrives, we will know the cost of setting up this connection for each wavelength. The wavelength that can be used to setup this connection with the least cost will be chosen. If there is more than one wavelength that can be used to setup this connection with the same least cost, the wavelength with the least number (FF) will be chosen.

3.5 SLSP-FF with Wavelength Converters

In Sections 3.2, 3.3, and 3.4, the network was assumed to have no wavelength converters. In this section, the SLSP routing algorithm, and FF wavelength assignment strategy are slightly enhanced so that they can take advantage of the wavelength converters in the network.

Before we get into the RWA algorithm, let us describe how the network is modeled. As described before, for a network with W available wavelengths and no wavelength converters, we can model the network as W identical single wavelength networks. Let consider the 4-node 3-wavelength network in Figure 3.8.

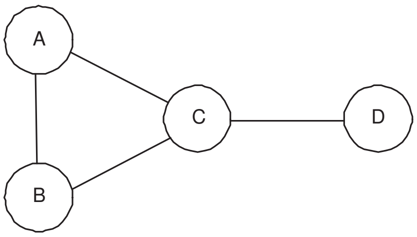

Figure 3.8: Network example.

In a network with no wavelength converters, the network in Figure 3.8 is modeled as 3 identical single wavelength networks, as in Figure 3.9.

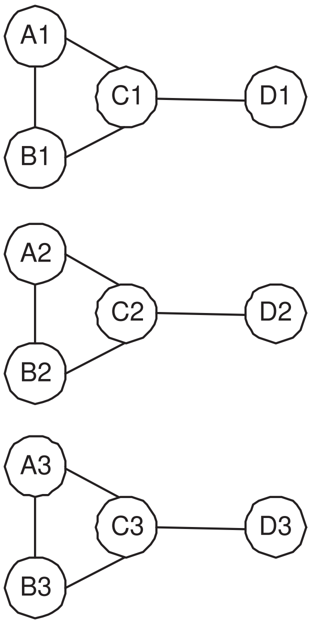

Figure 3.9: A 3-wavelength network is modeled as 3 identical single wavelength networks.

If a wavelength converter is introduced in a node, the nodes in these W single wavelength networks which represent this node will be joined. These edges have a fixed high cost, and they are always available (a connection can always uses these edges). A connection that uses one of these edges means a wavelength conversion is needed for this connection. Consider the same 4-node 3-wavelength network. Now, node C is equipped with a wavelength converter. Nodes C1, C2, and C3 in the first, second and third single wavelength network, respectively, are joined.

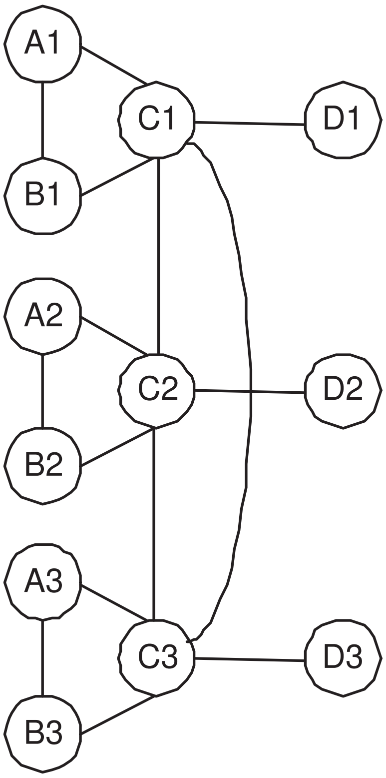

Figure 3.10: The introduction of a wavelength converter to a node joins the corresponding nodes in the single wavelength networks.

The connected network in Figure 3.10 can be treated as a single wavelength network. If a connection request between node A and node D (Figure 3.8) arrives, the source of the final connection can be either node A1, node A2 or node A3 in Figure 3.10. Similarly, the destination can be either node D1, node D2 or node D3. Any connection that connects node Ax to Dy, where $x = 1$, 2, or 3 and $y = 1$, 2, or 3, can be used as a connection between node A and D.

When using SLSP, in order to find a working path that uses the minimum number of wavelength conversions, an auxiliary graph is constructed. The construction of the auxiliary graph can be illustrated by the following SLSP-3 example. Consider the following 2-domain path (see Figure 3.11) chosen from one of the paths in the 3-shortest path list for a particular source and destination pair. In addition, along the intermediate nodes of this path, only $N1$ and $N3$ have wavelength converters.

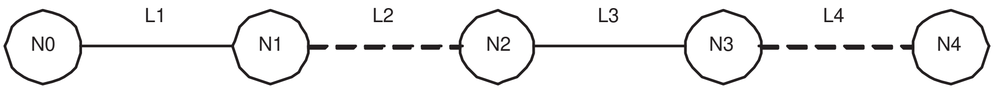

Figure 3.11: Example path from the 3-shortest path list.

By using SLSP-3, only $L1$ and $L3$ will be used in the final working path, while the connection between $N1$ and $N2$ and the connection between $N3$ and $N4$ have to be found adaptively. For a 2-domain path, the auxiliary graph is initialized as in Figure 3.12.

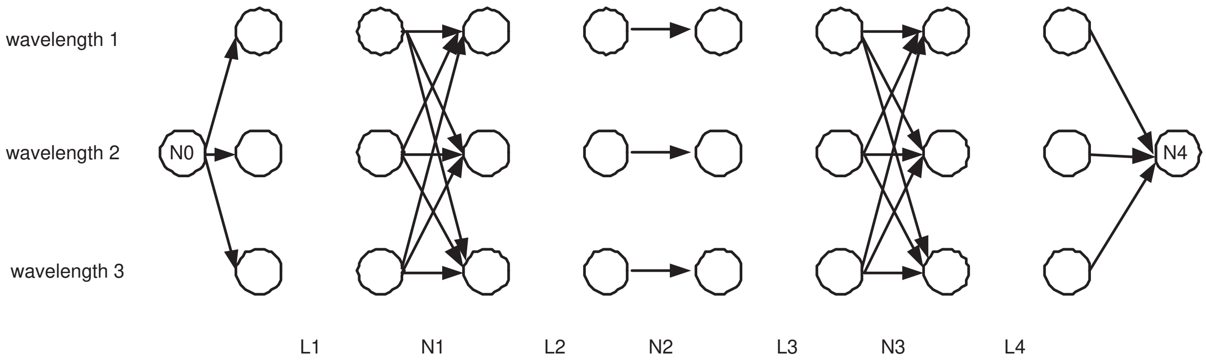

Figure 3.12: Auxiliary graph construction - step 1.

In Figure 3.12, since $N1$ and $N3$ have wavelength converters, the parts of the auxiliary graph that represent $N1$ and $N3$ form two complete bipartite graphs, while the part of the auxiliary graph that represents $N2$ only connects the nodes that belong to the same wavelength.

The edges that connect to $N0$ and $N4$ have weight 0. Moreover, edges that connect nodes in the same wavelength level also have weight 0. Other edges that connect nodes in different wavelength level have a high weight (300 in our simulation). The usage of these high weight edges means a connection has changed wavelengths along its path.

The next step for constructing this auxiliary graph is to update the information of $L1$ and $L3$, which will be included in the final working path if this path is chosen. For wavelength 1, if $L1$ is available, connect the two nodes representing $L1$ in wavelength 1, and give this edge a weight of 1. Otherwise, leave the two nodes unconnected. Repeat the same for wavelengths 2 and 3. Then repeat the whole thing for $L3$. For example, the auxiliary graph may now looks like Figure 3.13.

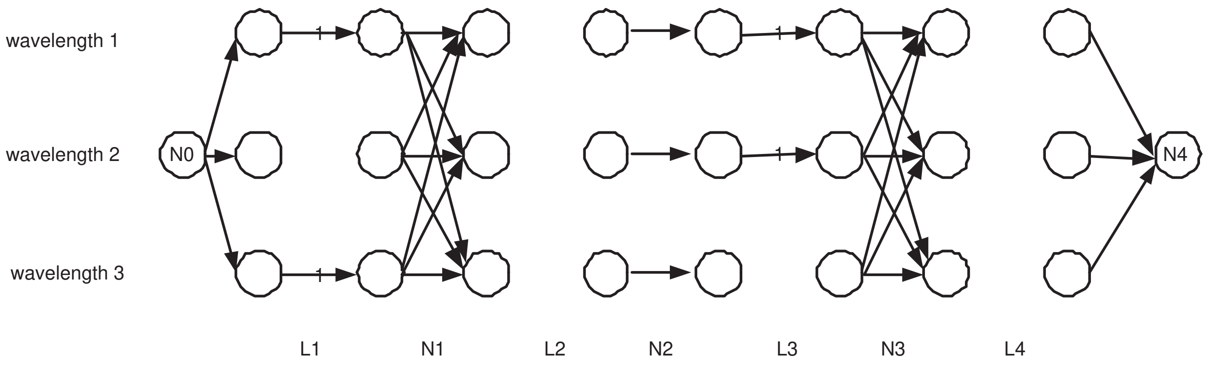

Figure 3.13: Auxiliary graph construction - step 2.

The next step for constructing this auxiliary graph is to update the information about the connection between $N1$ and $N2$ and the connection between $N3$ and $N4$, which have to be found adaptively.

For the connection between $N1$ and $N2$, try to establish a connection between $N1$ and $N2$ by using wavelengths 1 2 and 3. There are three cases. In one case, if a connection can be established by using wavelength 1 (2 or 3) only, connect the two nodes representing $L2$ in wavelength 1 (2 or 3), and set the weight of this edge to the cost of this adaptively-found connection. In the second case, if a connection exists that starts and ends at different wavelengths, connect the two nodes between $L2$ which represent the correct start and end wavelengths. For example, if a connection between $N1$ and $N2$ is found and this connection starts from wavelength 2 and ends at wavelength 3, connect the second node on the left of $L2$ to the third node on the right of $L2$, and set the weight of this edge to be the cost of this connection, which was found adaptively. In the third case, a connection between $N1$ and $N2$ cannot be found.

Assuming a connection between $N1$ and $N2$ exists, repeat the same for connections between $N3$ and $N4$. The auxiliary graph may now looks like Figure 3.14.

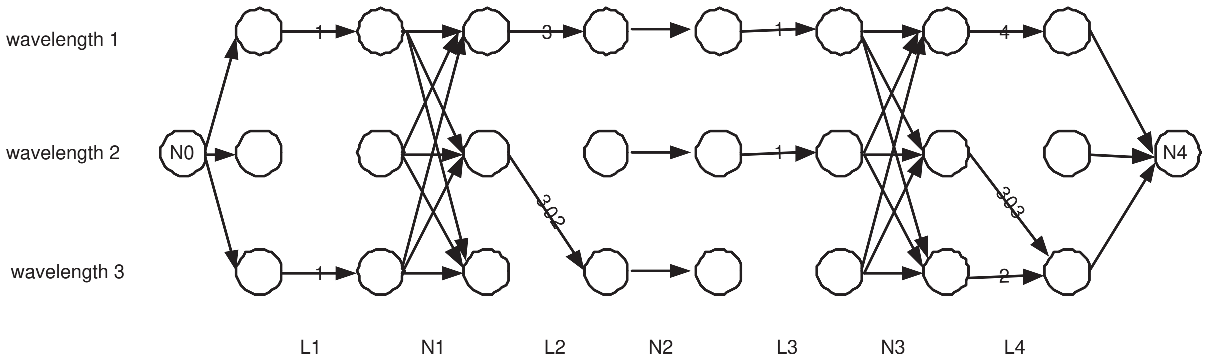

Figure 3.14: Auxiliary graph construction - step 3.

Now, the auxiliary graph is completed. By finding the shortest path between $N0$ and $N4$, the final working path for each domain and a temporarily assigned wavelength for each link could be found. In our example in Figure 3.14, the shortest path between $N0$ and $N4$ is obviously the route that uses only wavelength 1. By inspecting the resulting auxiliary graph link by link, we can deduce the final working path which uses the minimum number of wavelength conversions.

- $L1$: link $L1$ is used, and the temporarily assigned wavelength is wavelength 1.
- $N1$: no wavelength conversion is needed in node $N1$.
- $L2$: the 3-hop adaptively-found path between $N1$ and $N2$ is used, and the temporarily assigned wavelength is wavelength 1.
- $N2$: no wavelength conversion is needed in node $N2$. There is no wavelength converter in this node anyway.
- $L3$: link $L3$ is used, and the temporarily assigned wavelength is wavelength 1.
- $N3$: no wavelength conversion is needed in node $N3$.
- $L2$: the 4-hop adaptively-found path between $N3$ and $N4$ is used, and the temporarily assigned wavelength is wavelength 1.

After the working path is found, and a temporary wavelength is assigned for each link, the wavelength of each fragment may be reassigned to a lower numbered wavelength (FF). (For example, a path has two fragments if this path has changed wavelength once.) In our example, wavelength reassignment is not necessary because the only fragment of the connection is already using the lowest numbered wavelength. If a fragment, which comprises a set of links, is temporarily assigned wavelength x, where $x > 1$, the fragment will try using all lower numbered wavelengths (wavelength $x-1$, wavelength $x-2$, ... , wavelength 1). The lowest numbered available wavelength will be reassigned to this fragment.

Wavelength assignment for the working path has to be done before we can find protection paths because of the following. If wavelength converters are absent in the two end points of a P-domain, the protection path has to use the same wavelength as the working path at the two end points. Thus, the wavelength of the working path has to be determined before we can find the protection path for each domain. Consider the example in Figure 3.15.

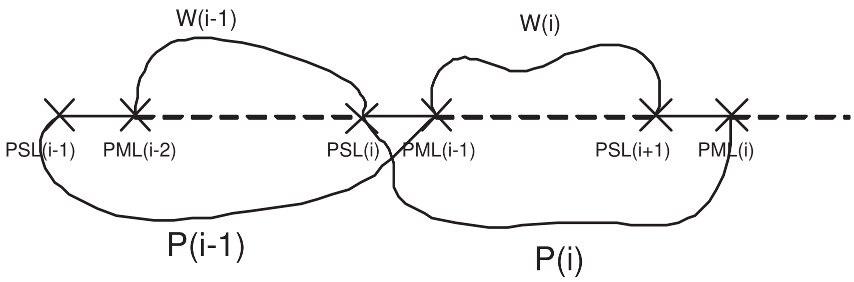

Figure 3.15: Wavelength assignment constraint for protection path.

If PSL(i) does not have a wavelength converter, $P(i)$ and $W(i-1)$ have to use the same wavelength at PSL(i). This is necessary because of the wavelength continuity constraint at PSL(i). When a protection switch occurs, traffic from $W(i-1)$ can be relayed to $P(i)$ using the same wavelength. By the same token, if PML(i-1) does not have a wavelength converter, $P(i-1)$ and $W(i)$ have to use the same wavelength at PML(i-1).

After the working path is found, we update the weight of each edge of the network by using the SRLG constraint. Then, the protection path for each domain could be found, given the above wavelength assignment constraint.

Chapter 4

Converter Placement Heuristics

The goal of all wavelength converter placement algorithms in dynamic traffic is to place a limited number of wavelength converters in the network such that the placement minimizes the overall blocking probability of the network. Finding an optimal placement of wavelength converters in a network with mesh topology is computationally unfeasible. Thus, sub-optimal placement strategies are often proposed.

One sub-optimal strategy for converter placement is to prioritize the nodes in the network. A node has a higher priority if when a full range wavelength converter is placed in this node, the benefit (decrease in blocking probability) will be higher than if the wavelength converter is placed in any lower priority node. In this project, we are only interested in using this sub-optimal strategy.

In general, the performance of a converter placement algorithm is measured in terms of blocking probability given the number of converters added. In this chapter, we introduce a novel converter placement algorithm, called Least Utilized Removed (LUR), for placing a limited number of wavelength converters in a survivable all-optical network. This algorithm uses the results of a sequence of simulations to prioritize the nodes in the network. To evaluate the performance of LUR, the nodes of a particular network were prioritized using LUR, and the performance (blocking probability given number of converters added) was tested via simulations. Then, an existing converter placement algorithm, called Minimum Blocking Probability First (MBPF) [8], which does not take network survivability into account was used to prioritize the nodes of the same network. Similarly, its performance was tested via simulations. The simulation results are reported in Chapter 5. In the following sections we describe the LUR and MBPF algorithms.

4.1 Least Utilized Removed (LUR)

This algorithm finds the node in the network with the least priority first.

Before the first simulation begins, equip every node in the network with a full range wavelength converter. After all the initialization is completed, simulate the network with a given traffic requirement. During the simulation, the utilization of each wavelength converter is recorded. After the simulation, the node which has done the least number of conversions (least utilized node) is selected as the node with the least priority. This converter is then removed from the network before the next simulation proceeds. The same simulation is performed except with the wavelength converter of the lowest priority node removed. After the second simulation, the node equipped with a wavelength converter which has done the least number of conversions is selected as the node with the second least priority. Then, as before, this converter is removed from the network before the next simulation proceeds. This process repeats until there is one converter left in the network. The node associated with this converter is the highest prioritized node.

In the evaluation of LUR, two different settings for the sequence of simulations used in LUR will be used. These settings will be described in Subsections 5.2.1, and 5.2.2. In Subsection 5.2.1, the normal SLSP survivable routing is used, and the prioritization of the nodes is based on the converter utilization by both working and protection paths generated by SLSP. In Subsection 5.2.2, a modified SLSP routing algorithm is used. Basically, the part of SLSP used to find a protection path for each domain is removed. Thus, the survivable property of SLSP is no longer in effect. The prioritization of the nodes is based on the converter utilization of the working traffic generated by this non-survivable version of SLSP.

4.2 Minimum Blocking Probability First (MBPF)

In dynamic traffic, the same converter placement algorithm performs quite differently under different RWA algorithms. Thus, when a converter placement algorithm is designed, the routing and/or wavelength assignment strategy used in the network should be taken into account [8]. MBPF is a converter placement algorithm that was designed to work with Fixed-Alternate Routing (FAR) algorithm. As described in Chapter 2, FAR pre-plans several fixed edge disjoint paths for each source and destination pair. When a connection request arrives, each pre-planned path will be checked to see whether resource reservations along each path is possible. The first available route will be chosen.

In the following subsections, an approximate mathematical model is described [8], which calculates the blocking probability of a network given the number and locations of wavelength converters. In MBPF, this mathematical model is used repeatedly in order to find the placement of a limited number of converters. MBPF places the converters one by one sequentially. For each iteration, MBPF tries to find the most important node from the candidate node list. (A candidate node is a node that does not have a wavelength converter.) A candidate node is considered most important, if placing a converter in that node will result in the most significant decrease in overall blocking probability.

Initially, there are no wavelength converters in the network, and all nodes are in the candidate node list. The goal is to find the most important candidate node. For an arbitrary candidate node, a wavelength converter is placed, and the overall blocking probability is calculated and recorded. Then, the wavelength converter is removed, and the same process is performed on another candidate node. After the same process has been performed on every candidate node, the one that results in the least overall blocking probability is chosen as the most important node. A wavelength converter is placed in that node permanently for subsequent calculation, and this node is removed from the candidate node list.

This procedure is repeated until there are no more wavelength converters or all nodes in the network are equipped with wavelength converters.

4.2.1 Approximate Model - System Parameters

- For each source and destination pair (s, d), connection requests arrive as a

Poisson process with rate λ.

- For each successfully established connection, the call-holding time is exponentially distributed with mean equal to 1. i.e. $\mu = 1$
- For each (s,d) pair, the offered traffic load $A^{(s,d)}$ is defined as the number of call requests per unit call-holding time.
- For each (s,d) pair, the carried traffic load $\overline{A^{(s,d)}}$ is defined as the number of accepted call requests per unit call-holding time.
- There are $M_{(s,d)}$ edge-disjoint paths provided for node pair (s,d). They are denoted by $R^1_{(s,d)}, R^2_{(s,d)}, ..., R^{M_{(s,d)}}_{(s,d)}$, in sequence.
- If there are $w^t_{(s,d)}$ wavelength converters on the path $R^t_{(s,d)}$, we can divide this path into $w^t_{(s,d)} + 1$ segments, where the kth segment is denoted by $R^{(t,k)}_{(s,d)}$.
- $B_{R^t_{(s,d)}}$ is the blocking probability of the path $R^t_{(s,d)}$.

4.2.2 Approximate Model - Blocking Probability Calculation Given the Number and Locations of Wavelength Converters

The overall blocking probability can be calculated by using $A^{(s,d)}$'s and $B_{R^t_{(s,d)}}$'s. Basically, the overall blocking probability is the ratio of the sum of all blocked traffic to the sum of all offered traffic.

$$P = \frac{\sum_{(s,d)}(A^{(s,d)} - \overline{A^{(s,d)}})}{\sum_{(s,d)} A^{(s,d)}} \tag{4.1}$$

By using FAR, a connection (s, d) is blocked only if it is blocked on all $M_{(s,d)}$ alternate paths. Moreover, these $M_{(s,d)}$ alternate paths are edge disjoint. In the approximate model, we assume that the blocking events of these paths are independent. We have

$$\overline{A^{(s,d)}} = A^{(s,d)} \left\{ 1 - \prod_{t=1}^{M_{(s,d)}} B_{R^t_{(s,d)}} \right\} \tag{4.2}$$

By substitution, we can express the blocking probability as follows:

$$P = \frac{\sum_{(s,d)} A^{(s,d)} \prod_{t=1}^{M_{(s,d)}} B_{R^t_{(s,d)}}}{\sum_{(s,d)} A^{(s,d)}} \tag{4.3}$$

4.2.3 Approximate Model - Iterative Approach for Calculating $B_{R^t_{(s,d)}}$'s

Now, the blocking probability is defined as a function of the $A^{(s,d)}$'s and $B_{R^t_{(s,d)}}$'s. The $A^{(s,d)}$'s are given as the traffic requirement. We only need to calculate $B_{R^t_{(s,d)}}$ to get the overall blocking probability.

The $B_{R^t_{(s,d)}}$'s are calculated by using an iterative approach.

Let us define $q_j(m_j)$ to be the probability that m_j wavelengths are free on link j. In the next two subsections, two sets of equations will be introduced.

- Set 1 - Given all $B_{R^t_{(s,d)}}$'s, calculate $q_j(m_j)$ for all links j.
- Set 2 - Given $q_j(m_j)$ for all links j, calculate all $B_{R^t_{(s,d)}}$'s.

By using Set 1 and Set 2, all the $B_{R^t_{(s,d)}}$'s can be calculated by using the following iterative approach:

1. Initialize all $B_{R^t_{(s,d)}}$'s to be zero, and all $q_j(m_j)$'s to be zero.
2. Use Set 1 to calculate all $q_j(m_j)$'s.
3. Use Set 2 to calculate all $B_{R^t_{(s,d)}}$'s.
4. If the new values of $B_{R^t_{(s,d)}}$'s have converged, the iteration is terminated. Otherwise, go back to step 2.

Note: convergence in step 4 means the absolute difference between the new values and the old values are less than some predefined small value.

4.2.4 Set 1 - Given All $B_{R^t_{(s,d)}}$'s, Calculate $q_j(m_j)$ for All Links j

Let X_j be the random variable that denotes the number of free wavelengths on link j. By the approximate model, we assume that X_i and X_j are independent , where i and j are different links. Moreover, assume the call requests arrival process on link j is a Poisson process with rate α_j.

First, we would like to calculate α_j for all links by using the $B_{R^t_{(s,d)}}$'s and $q_j(0)$'s from the previous iteration. (If it is the first iteration, use the initial values)

$$\alpha_j(1 - q_j(0)) = \sum_{(s,d)} \sum_{\substack{1 \le t \le M_{(s,d)} \\ j \in R^t_{(s,d)}}} A^{(s,d)} B_{R^1_{(s,d)}} B_{R^2_{(s,d)}} \cdots (1 - B_{R^t_{(s,d)}}) \tag{4.4}$$

Based on the model, the arriving and serving behavior of the links forms an $M/M/C/C$ system which can be described as the following Markov chain.

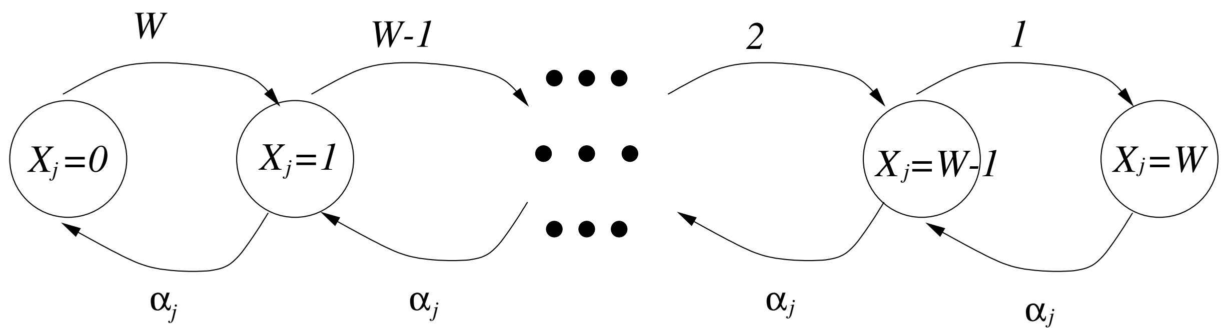

Figure 4.1: Markov chain for free wavelength distribution on link j.

By solving the Markov chain, we get

$$\begin{aligned} q_j(m_j) &= P(X_j = m_j) \\ &= \frac{\prod_{i=1}^{m_j}(W-i+1)}{\alpha_j^{m_j}} P(X_j = 0) \end{aligned} \tag{4.5}$$

where

$$P(X_j = 0) = q_j(0) = \left\{ 1 + \sum_{m_j=1}^{W} \frac{\prod_{i=1}^{m_j}(W-i+1)}{\alpha_j^{m_j}} \right\}^{-1} \tag{4.6}$$

4.2.5 Set 2 - Given $q_j(m_j)$ for All Links j, Calculate All $B_{R^t_{(s,d)}}$'s

Let define the term $u_i(m_j, R^{(t,k)}_{(s,d)})$

$$\begin{aligned} u_i(m_j, R^{(t,k)}_{(s,d)}) &= P(i \text{ common wavelengths are available on segment } R^{(t,k)}_{(s,d)} \mid \\ &\qquad m_j \text{ wavelengths are free on link } j \text{ in segment } R^{(t,k)}_{(s,d)}) \end{aligned} \tag{4.7}$$

where link j can be any link in the segment $R^{(t,k)}_{(s,d)}$.

In particular, $u_0(m_j, R^{(t,k)}_{(s,d)})$ is the probability that given m_j wavelengths are free on a link j of the segment $R^{(t,k)}_{(s,d)}$, there is no common free wavelength on the segment $R^{(t,k)}_{(s,d)}$.

If the total number of available wavelengths is W, the probability that there is at least one free common wavelength available on the segment $R^{(t,k)}_{(s,d)}$ is:

$$1 - \sum_{m_j=0}^{W} q_j(m_j) u_0(m_j, R^{(t,k)}_{(s,d)}) \tag{4.8}$$

With an approximate assumption that the blocking events on all the segments are independent, the blocking probability of any path $R^t_{(s,d)}$ can be derived as:

$$B_{R^t_{(s,d)}} = 1 - \prod_{k=1}^{w^t_a+1} \left\{ 1 - \sum_{m_j=0}^{W} q_j(m_j) u_0(m_j, R^{(t,k)}_{(s,d)}) \right\} \tag{4.9}$$

Now, we need to know how to calculate $u_i(m_j, R^{(t,k)}_{(s,d)})$.

Let define the term $p^h_i(m_{j_1}, m_{j_2}, m_{j_3}, \cdots, m_{j_h})$ for a h-hop segment.

$$\begin{aligned} p^h_i(m_{j_1}, m_{j_2}, \cdots, m_{j_h}) \quad = \quad & P(i \text{ common wavelengths available on the } h \text{ hop segment } | \\ & \text{link } j_1 \text{ in the segment has } m_{j_1} \text{ free wavelengths AND} \\ & \text{link } j_2 \text{ in the segment has } m_{j_2} \text{ free wavelengths AND} \\ & \text{link } j_3 \text{ in the segment has } m_{j_3} \text{ free wavelengths AND} \\ & \bullet \\ & \bullet \\ & \bullet \\ & \text{link } j_h \text{ in the segment has } m_{j_h} \text{ free wavelengths}) \end{aligned} \tag{4.10}$$

Note: the links j_1, j_2,...,j_h do not have to be in sequence.

Let the segment $R_{(s,d)}^{(t,k)}$ be an h-hop segment and assume that link j is link j_1. Then

$$u_i(m_j, R_{(s,d)}^{(t,k)}) = \sum_{m_{j_2}=0}^{W} \sum_{m_{j_3}=0}^{W} \cdots \sum_{m_{j_h}=0}^{W} \left\{ \prod_{l=2}^{h} q_{j_l}(m_{j_l}) \right\} \times p_i^h(m_j, m_{j_2}, m_{j_3}, \cdots, m_{j_h}) \quad (4.11)$$

Now, we need to know how to calculate $p_i^h(m_{j_1}, m_{j_2}, m_{j_3}, \cdots, m_{j_h})$.

For a h-hop segment, by conditioning on the number of common free wavelengths in an $(h-1)$ hop subsegment and by the law of total probability, the expression $p_i^h(m_{j_1}, m_{j_2}, \cdots, m_{j_h})$ can be calculated recursively as:

$$p_i^h(m_{j_1}, m_{j_2}, m_{j_3}, \cdots, m_{j_h}) = \sum_{k=0}^{W} p_i^2(k, m_{j_h}) p_k^{h-1}(m_{j_1}, m_{j_2}, \cdots, m_{j_{h-1}}) \quad (4.12)$$

where $p_i^2(x, y)$ is defined as the probability that i common wavelengths are free in a 2-hop segment, given one link has x free wavelengths, while the other has y free wavelengths.

Note that several of the terms in the sum in Equation (4.12) can be immediately be set to zero, depending on the values of $m_{j_1}, m_{j_2}, \cdots, m_{j_{h-1}}$; for example, terms for which $k > min(m_{j_1}, m_{j_2}, \cdots, m_{j_{h-1}})$ are zero.

$$p_i^2(x, y) = \begin{cases} \beta(x, y, i) & \text{if } x \geq y \geq i; x + y - i \leq W; 1 \leq x \leq W; 1 \leq y \leq W \\ \beta(y, x, i) & \text{if } y \geq x \geq i; x + y - i \leq W; 1 \leq x \leq W; 1 \leq y \leq W \\ 0 & \text{otherwise} \end{cases} \quad (4.13)$$

where

$$\beta(x, y, i) = \binom{y}{i} \left(\prod_{k=1}^{i} \frac{x - k + 1}{W - k + 1} \right) \left(\prod_{k=1}^{y-i} \frac{W - x - k + 1}{W - i - k + 1} \right) \quad (4.14)$$

To understand Equation (4.14), consider the following analogy [26]. There are W bins, x red balls, and y blue balls, where $x \geq y$. First, put the x red balls into the bins randomly (maximum 1 red ball per bin). Then, put the y blue balls into the bins randomly (maximum 1 blue ball per bin). The probability $\beta(x, y, i)$ is equivalent to the probability that i bins have two balls (one red ball and one blue ball).

4.3 Remarks on MBPF

From all the literature that I could find on converter placement, none of it takes network survivability into account. In [8], the study showed that the converter placement and RWA algorithms are closed related. Thus, when designing a converter placement algorithm, the routing and/or wavelength assignment strategy used in the network should be taken into account. Among the converter placement algorithms, MBPF was chosen to compare with the novel Least Utilized Removed converter placement algorithm because MBPF was designed for the FAR routing algorithm, and the routing algorithm that we used, SLSP, is similar to FAR.

For routing strategy, MBPF was designed for FAR, while the simulations in LUR use SLSP-3. The way that SLSP-3 finds a working path is very similar to that of FAR. FAR pre-plans several edge disjoint paths for each source and destination pair before the network is in operation. These pre-planned paths will be tried one by one when a connection request for a particular source and destination pair arrives. For SLSP-3, several paths, not necessarily edge disjoint, are pre-planned for each source and destination pair. These paths will be used to define the end points of some overlapping domains. If a particular pre-planned path is chosen to make a connection, for each domain, only the first edge in the pre-planned path will be used to construct the working path, while the rest will be found adaptively based on network conditions. The concatenation of working paths of each domain forms the final working path. In a sense, this mechanism is half-adaptive or half-static. It will perform slightly better than FAR, which is completely static. However, for SLSP-3, since each domain has a small diameter (3-hops), working on a network with a small diameter, like the 24-node "National Network" for the contiguous United States that we consider in Chapter 5 (the most separated node pair has a shortest path only 7 hops long), the performance gain would be small. Based on this analysis, we decided to use MBPF to compare with LUR(W+P) (see Section 5.2.1) because both of them were designed for similar routing algorithms.

Finally, there is one more remark for MBPF. MBPF is not very computationally feasible for a network with a large diameter and a large number of available wavelengths. The series of summations in Equation (4.11) makes it computationally heavy. For example, for a 10-hop path and 10 available wavelengths, there are 9^{11} additions to be computed. Moreover, the same equation has to be used for each iteration. Thus, it will take a lot of time before a sub-optimal converter placement can be found.

Chapter 5

Results and Discussion

This chapter presents the descriptions and results of the simulations that were performed for this project. As mentioned in the Project Overview in Chapter 1, this project has two goals. The first goal is to investigate the performance improvement of a survivable network when wavelength converters are introduced in every node. The second goal is to determine the placement of a limited number of wavelength converters in a survivable network that gives sub-optimal performance. The first goal is accomplished by using simulations described in Section 5.1, and the results will also be presented in this section. After the results are presented, the significance and some interesting aspects of the results will be discussed. The second goal of this project is accomplished by using the novel converter placement algorithm LUR. Its performance was tested via simulations described in Section 5.2. The results will also be presented in this section. Similarly, after the results are presented, the significance of the results and some aspects of the converter placement algorithms will be discussed. For details of the simulation programs, please refer to the Appendix.

5.1 Effects of Converters in Survivable Networks

5.1.1 Simulation Set 1 - No Conversion

This set of simulations investigates the performance of SLSP in networks with no wavelength converters in terms of blocking probability and unfairness.

Setting

The performance of the protection strategy, SLSP, was tested on a 24-node "National Network" for the contiguous United States [1] (pp.642) (see Figure 5.1). The network was assumed to have two fibers between adjacent nodes, one in each direction, and each fiber contains x wavelengths.

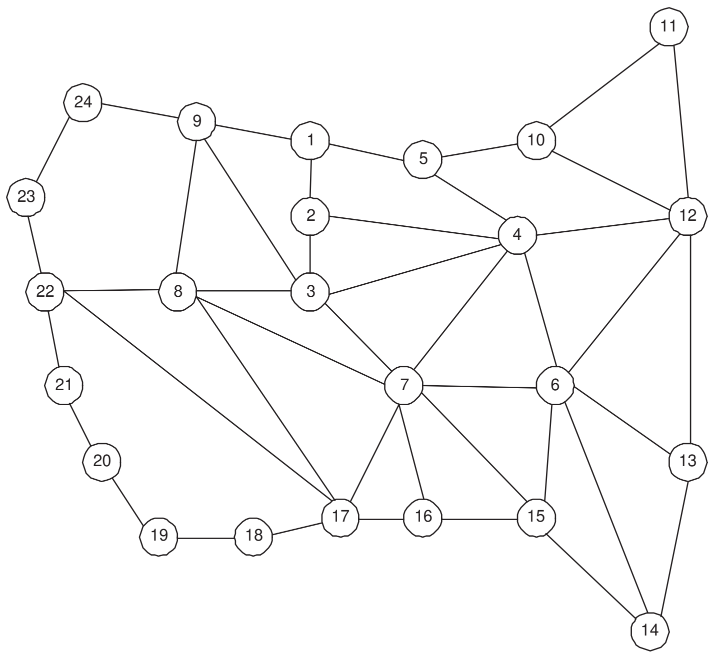

Figure 5.1: "National Network" for the contiguous United States.

The survivable routing strategy, SLSP-3, which means the diameter of each domain is 3 hops, was used. Moreover, for each source and destination pair, the 3-shortest paths were used to pre-plan the potential domains before the network was in operation. For wavelength assignment, as described in Section 3.4, the First Fit (FF) strategy was used.

The arrivals of connection requests for the whole network were taken to be a Poisson process with rate λ. When a connection request arrives, a source and destination pair was chosen randomly. The bandwidth required by each connection occupied an entire wavelength. If SLSP-3 could find a suitable working and protection path pair for each domain, the resources required by this connection would be reserved. Otherwise, the connection request was blocked. After a connection was successfully established, its departure time was assumed to be exponentially distributed with rate μ . When a connection departed, the resources used would be released.

In this set of simulations, the load was defined as $(\lambda/\mu)/\binom{n}{2}$ where n is the number of nodes in the network, and the unit of the load is Erlangs per node pair. It is different from the traditional definition of load which takes the number of available wavelengths into account. The load is decoupled from the number of available wavelengths because we would like to see the effect of different number of wavelengths used on blocking probability and unfairness given a fixed external traffic requirement.

The blocking probability was defined as follows:

$$P_b = \frac{\text{number of blocked connection requests}}{\text{total number of connection requests}} \tag{5.1}$$

The unfairness was defined as follows:

$$\overline{F} = \frac{\sum_{l=1}^{L}(P_b^l - P_b)^2}{L} \tag{5.2}$$

where

$$
\begin{aligned}
L &= \text{length of the longest shortest path in the network based on number of hops} \\
P_b^l &= \frac{\text{number of blocked connection requests with length } l}{\text{total number of connection requests with length } l}
\end{aligned}
$$

A length l connection means the shortest distance (number of hops) between the source and destination of this connection is l.

As described by Equation (5.2), it is considered maximally fair if all the P_b^l's are equal. In this case, the unfairness is 0. As the differences between the P_b^l's increase, the unfairness increases.

Simulations

Two sets of results were obtained from simulation set one. First, the relationship among the load, blocking probability and unfairness was investigated. The number of wavelengths used was fixed to be 16, and different loads (0.2, 0.35, 0.5, 0.75, and 1) were tested. The blocking probability and unfairness of each load were recorded, and data was derived by averaging the results of 3 trials. For each trial, the simulation generated at least 10,000 connection requests.

Second, the relationship among the number of available wavelengths, blocking probability and unfairness was investigated. The load was fixed at 0.75, and different numbers of available wavelengths (8, 12, 16, 20, and 24) were tested. The blocking probability and unfairness for each number of available wavelengths were recorded, and data was derived by averaging the results of 3 trials. For each trial, the simulation generated at least 10,000 connection requests.

5.1.2 Simulation Set 2 - Full Conversion

This set of simulations investigates the performance of SLSP in networks with full wavelength conversion capability. The blocking probability and unfairness were investigated.

Setting

The setting for simulation set two was almost exactly the same as simulation set one except there were full range wavelength converters in all nodes and there were two additional rules for routing and wavelength assignment.

1. Because the cascade of wavelength conversions will cause signal attenuation, the wavelength converters were treated as last resorts. In other words, a route which used a fewer number of wavelength conversions would cost less than one which used more wavelength conversions.

2. After a route is chosen, for each fragment (e.g., a path has two fragments if there is one wavelength conversion along this path), the least numbered available wavelength (FF) would be assigned for each fragment.

The modified version of SLSP-FF which takes advantage of wavelength converters in the network was described in Section 3.5.

Simulations

The simulations performed were exactly the same as the simulations described in Subsection 5.1.1.

5.1.3 Results - Blocking Probability and Unfairness Under Different Loads

Load	Blocking Prob(full conversion)	Blocking Prob(no conversion)
0.20	0.001795	0.030047
0.35	0.034980	0.115983
0.50	0.098680	0.183420
0.75	0.173867	0.286507
1.00	0.291033	0.336833

Table 5.1: Blocking probability versus load (16 wavelengths).

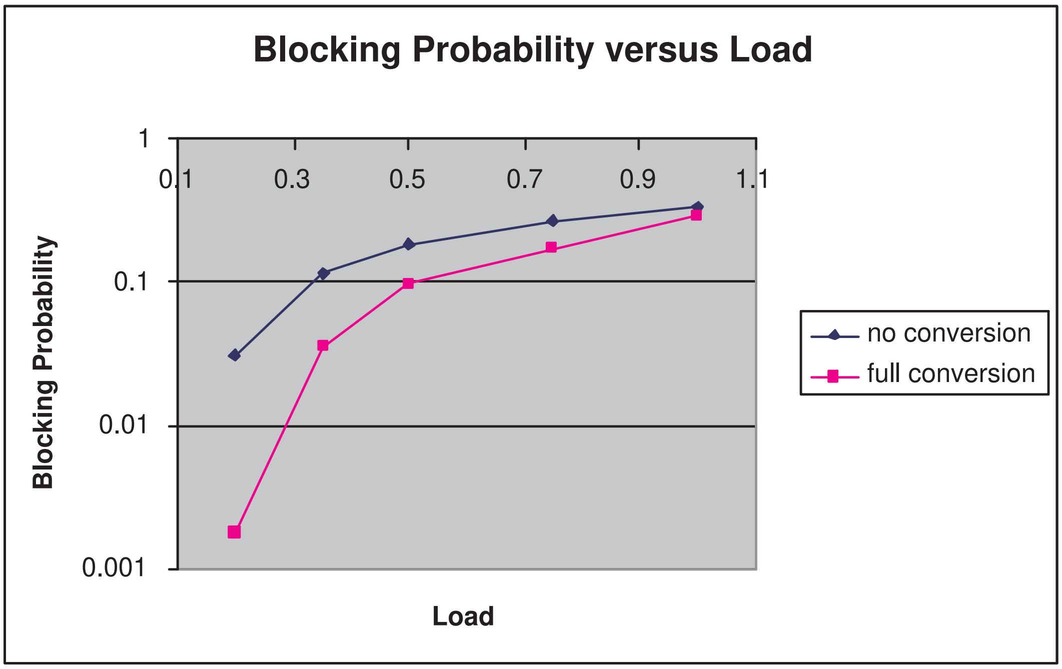

Figure 5.2: Blocking Probability versus load (16 wavelengths).

Due to the fact that wavelength converters can alleviate the burden of the wavelength continuity constraint, it was expected that the blocking probability would decrease when wavelength converters were introduced into the network. However, there is one point that is worthwhile to mention. As the load decreases, the significance of the wavelength converter increases. This can be explained by the following. In a high load situation, most of the blocking was caused by a lack of network resources since the network was fully loaded most of the time. Thus, the wavelength converter could not help much in this situation. In contrast, in a low load situation, most blocking was caused by lack of continuous free wavelengths along the paths of the connections. However, free capacity existed along the paths for most connections. Therefore, wavelength converters could help in solving this problem by changing the wavelength used by a connection during its journey from source to destination.

Load	Unfairness(full conversion)	Unfairness(no conversion)
0.20	0.00000707	0.013958
0.35	0.00211500	0.086683
0.50	0.01276500	0.131855
0.75	0.03090300	0.162231
1.00	0.05233500	0.171588

Table 5.2: Unfairness versus load (16 wavelengths).

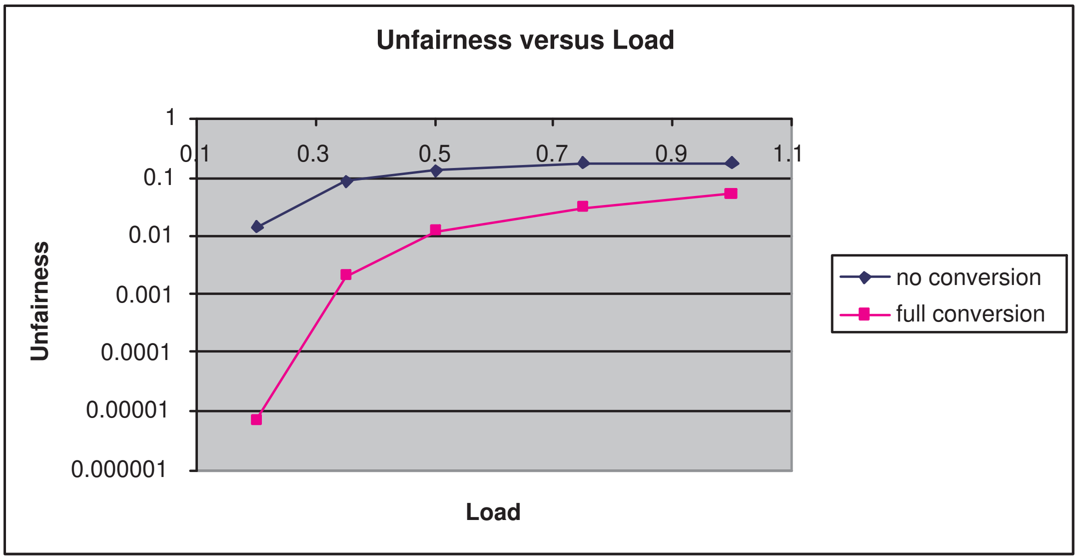

Figure 5.3: Unfairness versus load (16 wavelengths).

Similarly, it was expected that the unfairness would decrease when wavelength converters were introduced into the network. Unfairness exists because long connections require more network resources than short connections. Thus, it is more difficult to set up long connections. In turn, the blocking probabilities of longer connections are higher than those of shorter connections. Unfairness always exists in a network. However, the introduction of wavelength converters can reduce the unfairness in a network especially in a low load situation. Similar to the reasoning for blocking probability, in a high load situation, most of the blocking was caused by a lack of capacity. Even though wavelength converters were present, both blocking probability and unfairness showed only little improvement. In a low load situation, more network resources were not utilized. For a network with no wavelength converters, most of the blocking was caused by lack of a continuous free wavelength. It was harder to find a continuous free wavelength for long connections than for short connections. Thus, the blocking probabilities for short connections are relatively lower than those of long connections. This caused high unfairness. With the help of wavelength converters, the fragmented resources of the network could be used. This benefited long connections more than short connections. Thus, the blocking probabilities for long connections decreased, while the blocking probabilities for short connections remained almost unchanged. In turn, unfairness decreased.

5.1.4 Results - Blocking Probability and Unfairness Under Different Number of Available Wavelengths

# Wavelengths	Blocking Probability (full conversion)	Blocking Probability (no conversion)
8	0.503633	0.540587
12	0.313600	0.403480
16	0.173867	0.286507
20	0.105900	0.237130
24	0.054333	0.180333

Table 5.3: Blocking probability versus number of wavelengths (load=0.75).

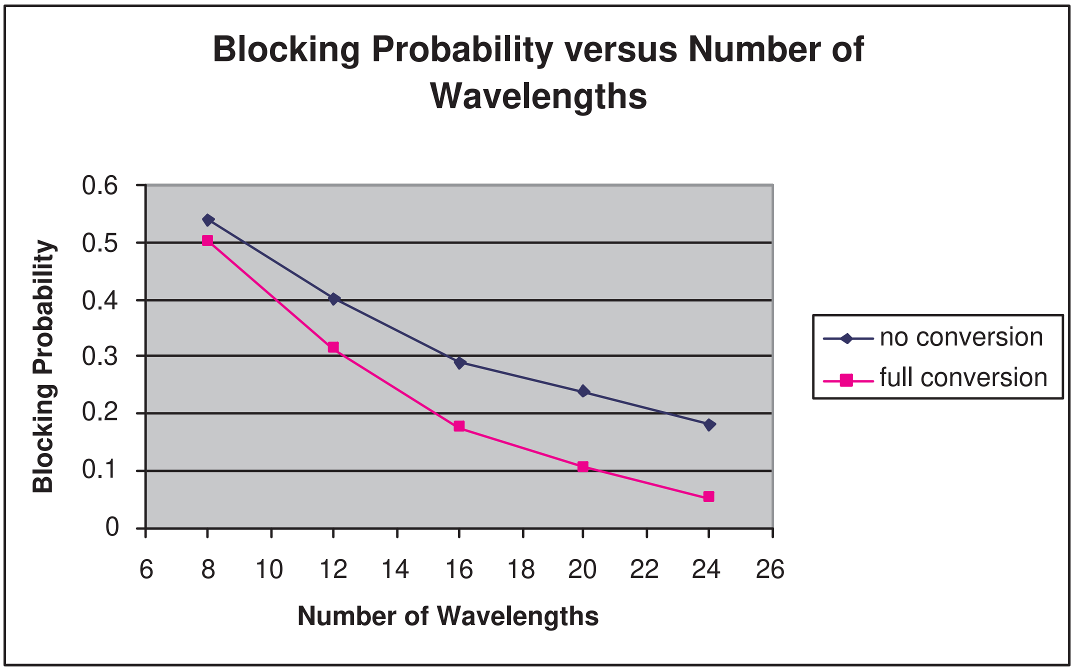

Figure 5.4: Blocking probability versus number of wavelengths (load=0.75).

Similar to the previous subsection, it was expected that the blocking probability would decrease when wavelength converters were introduced into the network. When the number of available wavelengths increases, the amount of network resources available also increases. The situation is equivalent to decreasing the load in the network. Thus, with a small number of available wavelengths, the situation is analogous to the high load situation in the previous subsection. Similarly, with a large number of available wavelengths, the situation is analogous to the low load situation in the previous subsection. The results are consistent with this analogy.

# Wavelengths	Unfairness(full conversion)	Unfairness(no conversion)
8	0.064035	0.128165
12	0.047552	0.153967
16	0.030903	0.162231
20	0.014632	0.143048
24	0.004580	0.096647

Table 5.4: Unfairness versus number of wavelengths (load=0.75).

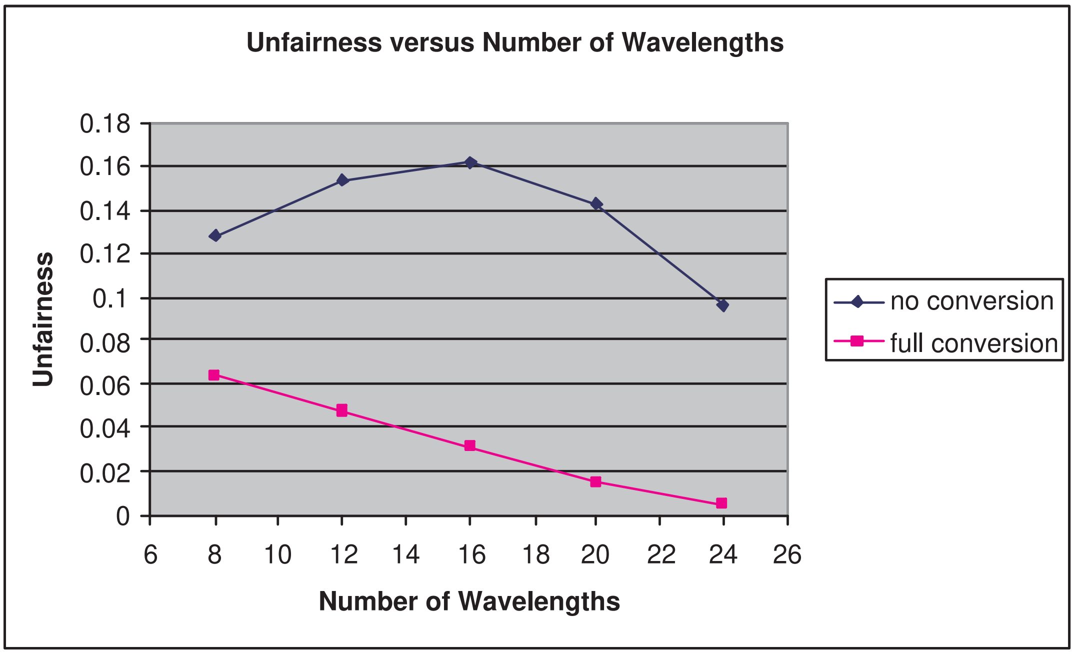

Figure 5.5: Unfairness versus number of wavelengths (load=0.75).

For the network with full wavelength conversion capability, the results for unfairness were expected. As the number of available wavelengths increases (analogous to decreasing the load), the unfairness decreases. For the network with no wavelength converters, the results were more interesting. The unfairness is maximized when the number of available wavelengths equals 16. This can be explained by the following.

When the number of available wavelengths was small, most of the blocking was caused by a lack of network resource. Most network resources were fully occupied most of the time; thus, it was difficult to setup both short and long connections. This caused low unfairness because blocking probabilities for both short and long connections were high. As the number of available wavelengths increased, the amount of network resources also increased. More short connections could take advantage of this increase of network resources than longer connections. This caused the blocking probability of short connections to become relatively lower than those of long connections. In turn, the unfairness increases. As the number of available wavelengths further increased, more network resources were available. In other words, more network resources were available most of the time; thus, it was easy to setup both short and long connections. This caused low unfairness because blocking probabilities for both short and long connections were low.

5.2 Heuristics for Converter Placement

In Section 4.1 the Least Utilized Removed (LUR) converter placement algorithm was introduced. By using the setting described in Subsections 5.2.1 (LUR(W+P)) and 5.2.2 (LUR(W)), two different node prioritizations for the same network were produced. In Section 4.2, another converter placement algorithm called Minimum Blocking Probability First (MBPF) was introduced. By using MBPF, another node prioritization for the same network was produced. The setting for MBPF is described in Subsection 5.2.3. In Section 5.2.4, simulation set 3 is described. This set of simulations evaluate the performance of the above three converter placement algorithms

(LUR(W+P), LUR(W), and MBPF). The performance of each converter placement algorithm was measured in terms of blocking probability given the number of converters added.

5.2.1 Node Prioritization Based on Converter Utilization of Both Working and Protection Traffic

This subsection describes the first setting for the simulations that was used in LUR. The setting was very similar to the setting described in Subsection 5.1.1. The network topology was chosen to be the same 24-node "National Network" for the contiguous United States in Figure 5.1. The network was assumed to have two fibers between adjacent nodes, one in each direction, and each fiber contains 8 wavelengths.

Again, the survivable routing strategy, SLSP-3, was used, and for each source and destination pair, the 3-shortest paths were used to pre-plan the potential domains before the network was in operation. By using SLSP-3, the path that used the least network resources (least cost) was chosen when a routing decision was made. In our simulation, the cost of using a wavelength converter was relatively high. In other words, the routes which used fewer (or no) wavelength conversions would be considered to have a lower cost than those which used more wavelength conversions.

For wavelength assignment, a wavelength has to be assigned for each fragment of the connection (e.g., a path has two fragments if there is a single wavelength conversion along this path). After a route is chosen based on least cost, the least numbered available wavelength (FF) would be assigned for each fragment.

Similar to Section 5.1, in this section, the load was defined as $(\lambda/\mu)/(\binom{n}{2} \times 2)$, where

n is the number of nodes, and the unit of the load is Erlangs per node pair per direction. Since λ is the arrival rate to the whole network, $1/(2\binom{n}{2})$ was included in the definition because there are $\binom{n}{2}$ node pairs, and each node pair has two directions. In other words, the load was defined as the arrival rate per unit connection hold time per node pair per direction. In our simulations, $load = 0.125$ was chosen. Before each simulation commenced, the utilization of each wavelength converter was initialized to be zero. During simulation, when a wavelength converter was used, either by a working or protection path, the utilization of this wavelength converter was incremented by one. After each simulation, the converter associated with the node which has the least count of utilization would be chosen as the next least-important node, and this converter would be removed from the network before the next simulation commenced.

5.2.2 Node Prioritization Based on Converter Utilization of Working Traffic Only

This subsection describes the second setting for the simulations that was used in LUR. The setting was very similar to the above subsection except for the following two modifications. First, the SLSP-3 algorithm was modified to a non-survivable version. The part of SLSP used to find a protection path for each domain was removed. The prioritization of the nodes was based on the converter utilization of the working traffic generated by this non-survivable version of SLSP. Second, the load was doubled. Because there were no protection resources allocated for each connection, the network should be able to handle more connections. Thus, the load was increased.

(Note, however, that for performance comparison, the node prioritization is tested in Section 5.2.4 on a network with load = 0.125 and with protection paths.)

5.2.3 Node Prioritization by Using MBPF

This subsection describes the setting and parameters used in our evaluation of MBPF on a survivable network. The network topology used was the same as in Figure 5.1. Similar to the setting for the LUR algorithm, the network was assumed to have two fibers between adjacent nodes, one in each direction, and each fiber contained 8 wavelengths.

By using MBPF, the 3 disjoint shortest paths were pre-calculated for the FAR algorithm. If only two disjoint paths existed for a source and destination pair, only these 2 disjoint shortest paths were pre-calculated and used in the FAR algorithm.

In the setting of the LUR(W+P) algorithm, the load was 0.125. In order to make the setting as similar to that of LUR as possible, the arrival rate of each (s, d) pair was set to be 0.125. The offered traffic load, $A^{(s,d)}$, was defined as the number of call requests per unit call-holding time, and we assumed uniform traffic. Hence, for all (s, d) pairs, $A^{(s,d)} = 0.125$.

5.2.4 Simulation set 3 - Evaluation of Each Converter Placement Algorithm

In Section 4.1, LUR was introduced. By using the settings described in Subsections 5.2.1 and 5.2.2, two different node prioritizations were produced. By using MBPF, introduced in Section 4.2, a third node prioritization was produced. By using these three node prioritizations, the performance of each wavelength converter placement algorithm was tested.

Priority	Node (LUR(W+P))	Node (LUR(W))	Node (MBPF)
1	7	17	4
2	17	8	17
3	22	7	8
4	8	22	7
5	4	4	1
6	9	9	22
7	3	3	6
8	19	18	3
9	18	6	19
10	21	21	21
11	20	1	10
12	6	5	9
13	1	12	12
14	23	19	18
15	24	20	20
16	5	2	5
17	12	23	24
18	2	24	23
19	10	10	15
20	16	16	2
21	15	15	16
22	11	11	13
23	13	13	14
24	14	14	11

Table 5.5: Node prioritizations based on different converter placement algorithms.

For each node prioritization, wavelength converters were added to the network one by one based on their priorities. After each wavelength converter was added, a simulation was performed for 3 trials to determined the blocking probability given the number of wavelength converters added. The setting of the simulation is exactly the same as the setting described in Subsection 5.2.1. For each simulation, at least 200,000 connection requests were generated.

5.2.5 Results - Performance of the 3 Converter Placement Algorithms

# Converters	P_b (LUR(W+P))	P_b (LUR(W))	P_b (MBPF)
0	0.197573	0.197573	0.197573
1	0.189270	0.192412	0.189665
2	0.183327	0.185275	0.183478
3	0.175520	0.179208	0.177418
4	0.169593	0.169593	0.172473
5	0.164823	0.164823	0.169440
6	0.159525	0.159525	0.162730
7	0.155638	0.155638	0.161462
8	0.154370	0.155428	0.157625
9	0.153303	0.154617	0.156537
10	0.152148	0.154223	0.155323
11	0.150398	0.152658	0.155048
12	0.148423	0.150405	0.150568
15	0.142448	0.144518	0.144658
18	0.137522	0.137522	0.138503
24	0.135500	0.135500	0.135500

Table 5.6: Blocking probability given number of converters based on different converter placement algorithms.

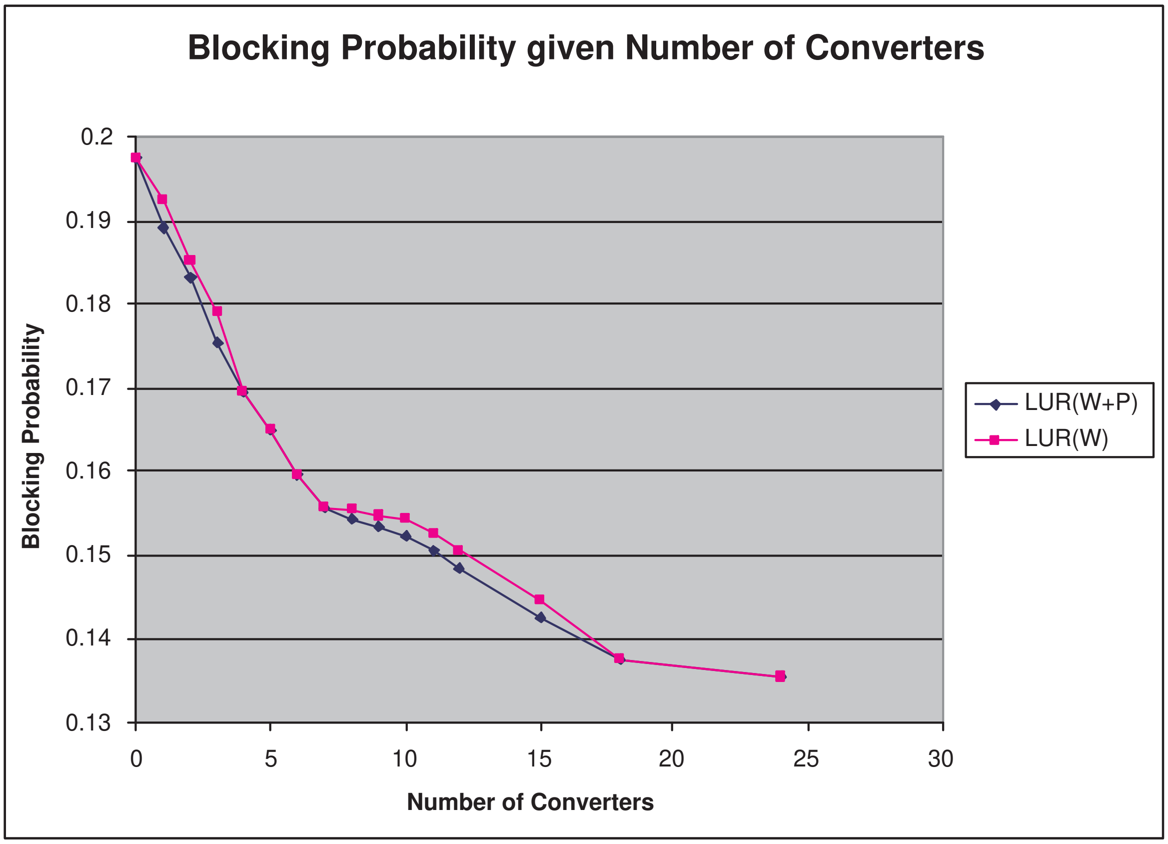

Figure 5.6: Performance difference between LUR(W+P) and LUR(W).

When comparing the performance of LUR(W+P) and LUR(W), the performance of LUR (W+P) is slightly better. This result suggests that the utilization of converters by protection traffic is an important factor in determining the priority of a node. Thus, when designing a converter placement algorithm for a survivable network, the utilization of converters by protection traffic has to be taken into account.

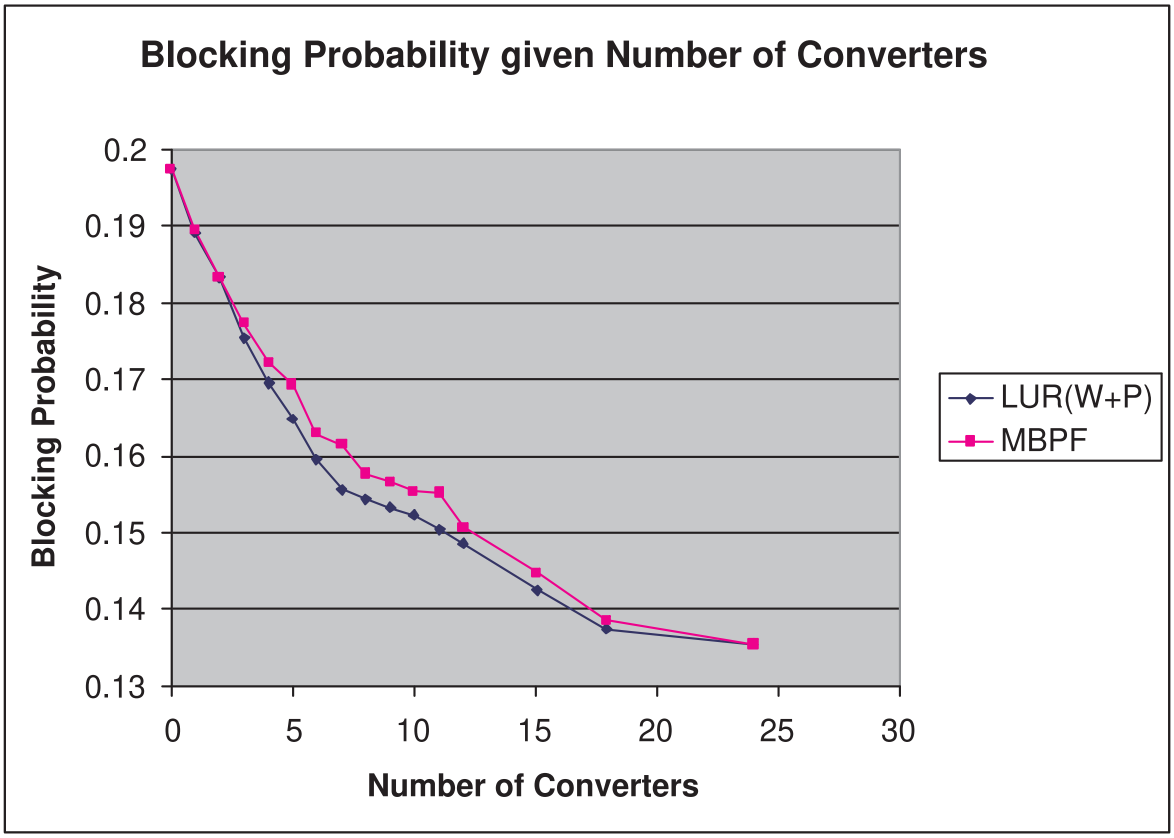

Figure 5.7: Performance difference between LUR(W+P) and MBPF.

When comparing the performance of LUR (W+P) and MBPF, the performance of LUR (W+P) is, again, slightly better. However, some advantages to using the LUR(W+P) algorithm over the MBPF algorithm are apparent. For instance, a blocking probability of ≈ 0.155 can be achieved with 3 fewer converters using LUR(W+P) instead of MBPF in our example. This result suggests that the resources used (path) by protection traffic is an important factor in determining the priority of a node. Thus, when designing a converter placement algorithm for survivable network, knowledge of the protection paths has to be noted.

Chapter 6

Conclusions

6.1 Discussion of Contributions

As mentioned in the Project Overview in Chapter 1, this project has two goals. The first goal is to investigate the performance improvement of a survivable network when wavelength converters are introduced in every node. Intuitively, the performance will increase when wavelength converters are introduced into the network. It is due to the fact that wavelength converters alleviate the burden of the wavelength continuity constraint and allow connections to used fragmented resources. The results presented in Chapter 5 further re-enforce this argument. The simulations that we have performed suggest that the introduction of wavelength converter will decrease the blocking probability and unfairness of a survivable network with dynamic traffic.

The second goal of this project is to investigate a sub-optimal placement of a limited number of wavelength converters in a survivable network. A converter placement algorithm called Least Utilized Removed (LUR) was developed for survivable networks.

LUR performed reasonably well in the simulations when converter utilization by protection traffic was taken into account. It slightly outperformed MBPF, which does not take network survivability into consideration when converter placement decisions were made. The results in Chapter 5 suggest that a converter placement algorithm designed for non-survivable networks would perform less efficiently in a survivable network. Thus, when algorithm designers design a converter placement algorithm for survivable networks, the effect of protection traffic has to be considered in the algorithm.

6.2 Future Work

LUR determines placement for a limited number of converters by using a sequence of simulations. For each simulation, the least important node is excluded from the potential converter recipient list. Thus, it takes a long time to determine a placement for a small number of available wavelength converters in a large network. If there are n nodes and 1 available wavelength converter, $n-1$ simulations have to be performed before we can find a sub-optimal location to place this converter.

Because of this time consuming nature, a reverse version of LUR may be developed and tested. In a reverse version of LUR, the first simulation will simulate the network with no converters. After this initial simulation, a score will be given to each node with no converter. This score should reflect the number of blocked connections that could be saved if there were a wavelength converter in this node. After the first simulation, a wavelength converter will be placed in the node with the highest score. This process repeats until all the available wavelength converters are used. Intuitively, this algorithm should perform well compared to LUR. However, a more rigorous definition of the score has to be made, and the performance of this algorithm has to be tested.

In general, simulations are very time consuming. It would be nice to develop an algorithm that is computationally efficient and does not depend on simulation. An approximate mathematical model, like the one in MBPF, may be developed to calculate the overall blocking probability of a survivable network given the number of converters added. One way to start this is to modify the approximate model in MBPF so that it can take network survivability into account. It is hard to modify the approximate model so that it can work with an advanced survivable routing strategy like SLSP in one step. It is relatively easy to modify it so that it can work with a simple survivable routing strategy like dedicated survivable FAR. Dedicated survivable FAR pre-plans several edge disjoint paths off line like the ordinary FAR. When a connection request arrives, one of the pre-planned paths will be used as the working path, while another path from the pre-planned path list will be selected as the dedicated (not shared) protection path. After this, the approximate model can be modified to work with an adaptive version of the dedicated survivable routing algorithm in which both working and protection paths are found adaptively based on network condition. Then, the approximate model can be further modified to work with advanced shared protection routing algorithms like ITSA and SLSP. It is reasonable to believe that using an approximate model to determine a converter placement will be more efficient than simulation-based algorithms in terms of computational time and performance.

Finally, there is one remark for the future work. Since the converter placement is an off-line problem, a long computational time may be acceptable. However, by using an efficient mathematical model, the computational time may be reduced by one or two orders of magnitude, thus enabling a network designer to run many converter placement scenarios, which may be desirable when designing the network topology, for example.

Appendix A

Description of Simulator

The purpose of this appendix is to explain the logic behind the simulation code.

Due to the fact that the topology that we used was always the same, the network topology was hard coded. Moreover, the arrival rate, departure rate, number of available wavelengths, number of connection requests generated by the simulation program and the locations of wavelength converters were also hard coded. However, they could be modified very easily. The only input for the program was the 3-shortest paths for each source and destination pair. Another program that implemented Yen's algorithm outputted the 3-shortest paths for each source and destination pair in the output window. These outputs were then saved as a text file, and this text file became the input of the simulation program.

The simulation was event driven in nature. A random number generator was used to create a Poisson arrival stream at a specified rate for each connection request, and it also generated the exponential holding time of each successful connection request. The following flow chart explains the flow of the simulation program.

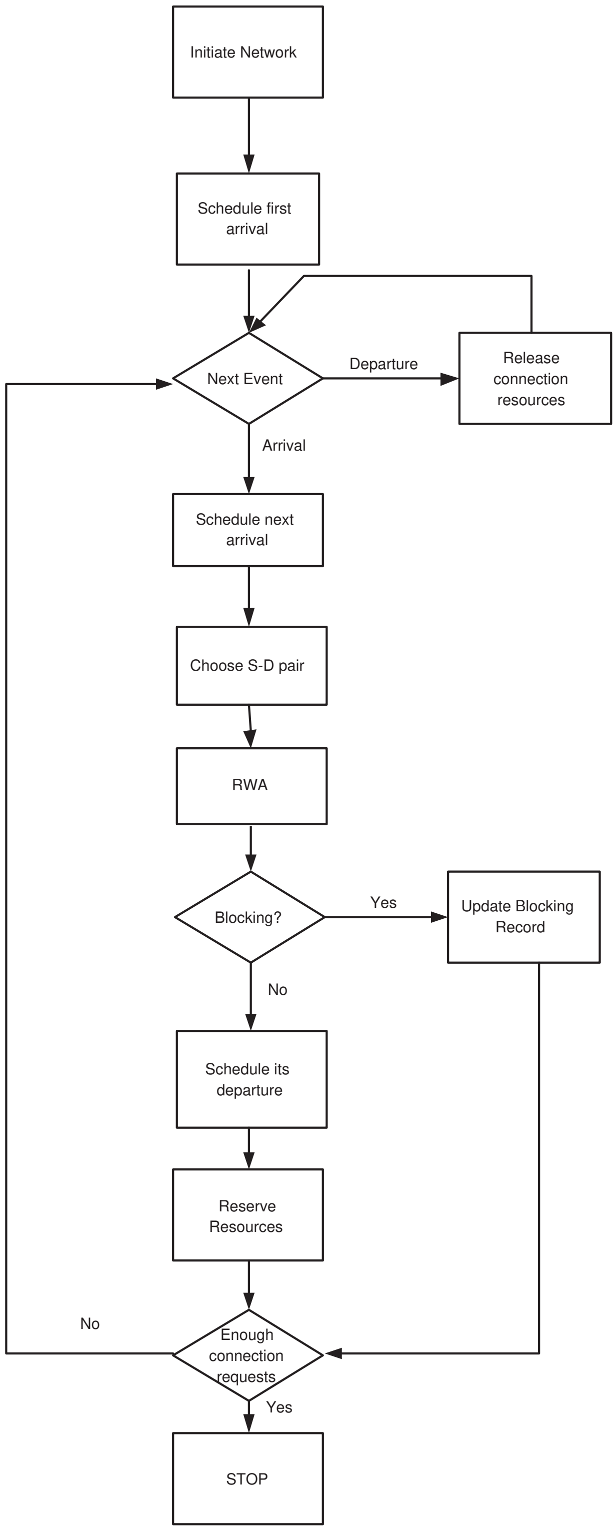

Figure A.1: Event driven simulator.

Now consider the flow of the RWA algorithm.

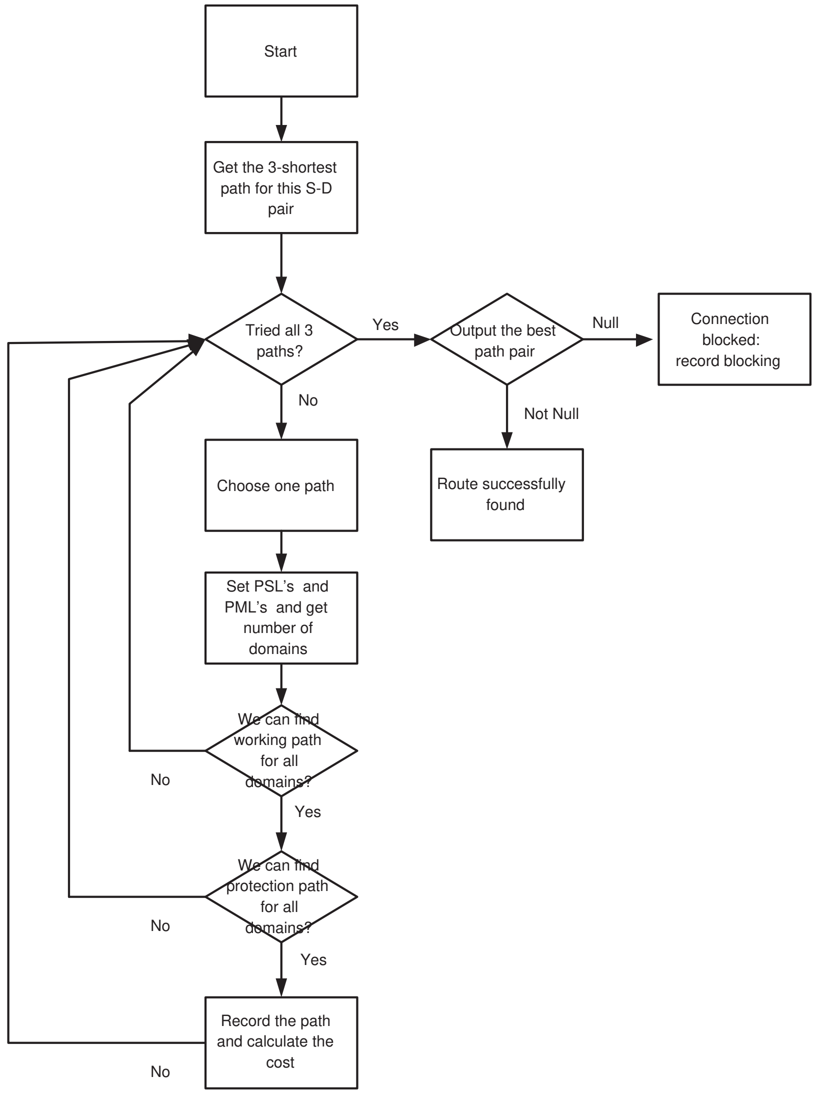

Figure A.2: RWA algorithm used in the simulation program.

Finally, there is one more remark for the simulation program. In Sections 5.1.1 and 5.1.2, when the effect of full wavelength conversion was investigated, accuracy of the data was not that important. We were only interested in looking for the trend. Thus, a fixed number of connection requests was used to find the blocking probability and unfairness. In Section 5.2.4, the 3 converter placement algorithms were investigated. The performance difference between these algorithms were small. Thus, accuracy of the results (blocking probability given the number of wavelength converters) was an important issue. For this reason, an additional piece of code was added to the simulation program. Basically, this additional code made the program output the result when the number of connection requests reached 200,000. After this, the simulation program will keep running, and output the blocking probability after every 1000 additional connection requests. It was then up to the user to realize when the simulation had converged to the true blocking probability by monitoring the output. In other words, the user decided when the simulation could be stopped once the blocking probabilities no longer changed (to the third significance digit in this case) from one output to the next.

Bibliography

[1] T.E. Stern, and K. Bala, *Multiwavelength Optical Networks A Layered Approach*, Addison-Wesley 1999.

[2] K.R.Venugopal, M. Shivakumar and P. S. Kumar, "A Heuristic for Placement of Limited Range Wavelength Converters in All-Optical Networks",*IEEE INFOCOM '99 Proceedings*, New York, NY, USA, March 1999, pp.908-915.

[3] S. Subramaniam, M. Azizoglu, and A. Somani, "All-Optical Networks with Sparse Wavelength Conversion", *IEEE Transactions on Networking*, Vol. 4, no.4, August 1996, pp.544-557.

[4] P.H. Ho and H.T. Mouftah, "A Framework of a Survivable Optical Internet using Short Leap Shared Protection (SLSP)", IEEE Workshop on High Performance Switching and Routing (HPSR 2001), Dallas, TX, USA, May 2001, pp.132-137.

[5] G. Xiao and Y.W. Leung, "Algorithms for Allocating Wavelength Converters in All-Optical Networks", *IEEE/ACM Transaction on Networking*, Vol. 7, no.4, August 1999, pp.545-557.

[6] S. Subramaniam, M. Azizoglu, and A.K. Somani, " On Optimal Converter Placement in Wavelength-Routed Networks", *IEEE/ACM transactions on Networking* Vol.7, no.5 October 1999, pp.754-766.

[7] S. Thiagarajan and A.K. Somani, " An Efficient Algorithm for Optimal Wavelength Converter Placement on Wavelength-Routed Networks with Arbitrary Topologies", *IEEE INFOCOM '99 Proceedings*, New York, NY, USA, March 1999, pp.916923.

[8] X. Chu, B. Li and I. Chlamtac, " Wavelength Converter Placement Under Different RWA Algorithms in Wavelength-Routed All-Optical Networks", *IEEE Transactions on Communications*, Vol.51, no.4, April 2003, pp.607-617.

[9] R. Ramamurthy and B. Mukherjee, "Fixed-Alternate Routing and Wavelength Conversion in Wavelength-Routed Optical Networks", *IEEE/ACM Transactions on Networking*, Vol.10. no. 3, June 2002, pp.351-367.

[10] K. Bala, T.E. Stern, and K.Bala. "A Minimum Interference Routing Algorithm for a Linear Lightwave Network", *IEEE GLOBECOM '91 Proceedings*, Phoenix, AZ, USA, December 1991, pp.1264-1269.

[11] D.Bertsekas and R. Gallager, *Data Network*, 2nd Edition, Prentice Hall, New Jersey, 1992.

[12] A. Sridharan and K.N. Sivarajan, "Blocking in All-Optical Networks", *IEEE INFOCOM '00 Proceedings*, Tel-Aviv, Israel, March 2000, pp.962-969.

[13] A.Mokhtar, *Routing and Dimensioning in Circuit-Switched Networks*, Reading, MA:Addison-Wesley, 1990.

[14] A. Mokhtar and M.Azizoglu, "Adapticve Wavelength Routing in All-Optical Networks", *IEEE/ACM Transactions on Networking*, Vol.6, no.2, April 1998, pp.197-206.

[15] P.A. Bonenfant and C.M.C. Davenport, "SONET Optical Layer Protection Switching", Bell Communications Research, Technical Report No. TM-24609, December 1994.

[16] P.H Ho and H.T. Mouftah, "Allocation of Protection Domains in Dynamic WDM Mesh Networks", *10th IEEE International Conference on Network Protocols Proceedings*, Paris, France, November 2002, pp.188-189.

[17] E. Modiano, A. Narula-Tam, "Survivalbe Lightpath Routing: A New Approach to the Design of WDM-Based Networks", *IEEE Journal on Selected Areas in Communications*, Vol.20, no.4, May 2002, pp. 800809.

[18] A. Sen, B. Hao, B.H. Shen and G. Lin, "Survivable Routing in WDM Networks - Logical Ring in Arbitrary Physical Topology", *IEEE International Conference on Communications, 2002*, New York, NY, USA, April 28 - May 2 2002, pp.2771 -2775.

[19] D. Li and X. Jia, "Placement of Wavelength Converters in Shared-per-link Structure in WDM Networks", *International Conference on Parallel Processing Workshops, 2001*, Valencia, Spain, 3-7 September 2001 , pp.277-282.

[20] K. Park, Y.S. Shin and S.M. Lee, "Wavelength Converter Location and Optical Fiber Dimensioning for Limited Channel Convertible Optical Networks",*IEEE*

GLOBECOM '98 Proceedings, Sydney, Australia, 8-12 November 1998, pp.2351 -2357.

[21] P.J. Wan, L. Liu, and O.Frieder, "Optimal Placement of Wavelength Converters in Trees and Trees of Rings", *Eight International Conference on Computer Communications and Networks, 1999, Proceedings* Natick, Massachusetts, USA, 11-13 October 1999, pp.392 -397.

[22] P. Wilfong, P. Winkler, "Ring Routing and Wavelength Translation", *9th ACM-SIAM Symp. on Discrete Algorithms Proceedings*,San Francisco, CA, USA, January 1998, pp.pp. 333-341.

[23] J. Kleinberg, A. Kumar, "Wavelength Conversion in Optical Networks", *10th ACM-SIAM Symposium on Discrete Algorithms Proceedings*, Baltimore, Maryland, USA, January 1999, pp.566-575.

[24] S.S. Skiena, *The algorithm Design Manual CD-ROM*, Springer-Verlag, New York, 1997.

[25] J.Y. Yen, "Finding the K-Shortest Loop-less Paths in a Network", *Management Science* , Vol. 17, 1971, pp.712-716.

[26] A. Birman, " Computing Approximate Blocking Probabilities for a Class of All-Optical Networks", *IEEE Journal on Selected Areas in Communications* Vol.14, no.5 June 1996, pp.852-857.

Vita

Name	Aaron Saifun So
Date of birth	January 22nd, 1978
Place of birth	Windsor, Ontario, Canada
Education	1. Master of Science (*Engineering*) Mathematics and Engineering (Communications) Queen's University, Kingston, Ontario, Canada, 2002-2003
	2. Bachelor of Science (*1st Class*) Mathematics and Engineering (Computing and Communications) Queen's University, Kingston, Ontario, Canada, 1998-2002
	3. Bachelor of Science (*with Distinction*) Life Science Queen's University, Kingston, Ontario, Canada, 1996-2001
Awards	R.S. McLaughlin Fellowship, 2002-2003
	Deans Award, 1999-2000,2001-2002
	Deans Scholar, 1999-2001
	Queens Appeal Undergraduate Scholarship, 1997-1998
	Annie-Lillie-Bentley Prize, 1997-1998

	Day Prize in Physics and Mathematics, 1997-1998
	Placed on Deans Honour List, 1997-1998
Experience	Software Designer (Intern), Optical Networks
	Nortel Networks, 2000-2001