1999

Designing a testbed for all-optical networks.

Tony Lai Ho. Lam

University of Windsor

Designing a Testbed For All-Optical Networks

by

Tony Lai Ho Lam

A Thesis
Submitted to the College of Graduate Studies and Research
through the School of Computer Science in Partial
Fulfillment of the Requirements for the Degree of
Master of Science at the
University of Windsor

Windsor, Ontario, Canada
1999

0-612-52592-9

Abstract

After a brief introduction to all-optical networks, including the structure of fiber optic network and the components in the network, we present a testbed for all-optical networks in this thesis. The testbed can evaluate the effect of wavelength conversion on different topologies in all-optical networks by studying the performance of a network in terms of the blocking probability. We are able to use the testbed to handle both regular and irregular topologies, accept different wavelength conversion schemes and allow the users to specify their own routing schemes. In this thesis, we have utilized the testbed to simulate a number of different networks. Examples that have been studied include the Ring, the De Bruijn graph, the Manhattan Street Network and an arbitrary irregular topology. Experimental results and some critical conclusions are included in this thesis.

We have implemented the testbed using C and Java native methods.

In loving memory of my father

Acknowledgments

First of all, I would like to acknowledge my sincere gratitude towards my supervisor Dr. S. Bandyopadhyay. He bore with me through the most difficult time of my graduate study. He was always there for me when I needed him. I am proud to be one of his students. Also, I would like to thank my committee members Dr. I. Tjandra and Dr. H. Kwan for their valuable comments and suggestions regarding the thesis. All my colleagues in university have helped me throughout my school years. I feel very fortunate and honored to be in this university.

My family has been a constant support for when I felt frustrated and lost. I cannot imagine how I could have done this thesis without their encouragement. Therefore I dedicate this piece of my work to my family, especially in the memory of my father.

Finally, I would like to thank all my brothers and sisters in church. They prayed for me and this thesis from the beginning to the end. May God bless all of us.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1 INTRODUCTION

A fiber optic network communicates information in the form of optical signals in a very fast and reliable way. Telecommunication nowadays is going through a large-scale transformation. Users demand that they have the ability to send huge amounts of information almost instantly [10]. For example, the World-Wide-Web, with multimedia features, is already used widely and this will increase greatly in the next century. Also, there is a rapid increase in communication need throughout the industry. Optical networks seem to provide a promising future for us by providing:

1. Huge bandwidth
2. Protocol transparency
3. High reliability
4. Simplified operation and management

Among various multiplexing schemes, wavelength division multiplexing (WDM) is a way to provide concurrency in a multiwavelength network [12]. In a WDM network, one of the concepts that has been proposed to utilize the huge bandwidth of fibers is wavelength conversion.

Investigators are currently looking at a variety of networks and examining their performance with respect to a number of metrics. These studies are being carried out independently and each study involves the development of complex simulation software. To our knowledge, there is no general-purpose simulation software for all-optical single-hop networks.

Therefore, this thesis reports our work on the development of a testbed for single-hop all-optical networks.

1.1 Motivation and Objectives of This Investigation

Our intention is to utilize the common features of different architectures to test our hypothesis using the testbed we have developed. In any network, when a communication is requested, the attempt to establish a communication may or may not succeed. If an attempt to establish a communication does not succeed, the communication is said to be "blocked" and may have to be attempted later. The quality of service from a network may be characterised in terms of the blocking probability, defined as the ratio of the number of blocked communication to the total number of attempted communication. It is expected that a "good" network has a very low blocking probability. In other words, most attempts for communication will succeed. When we compare different architectures, the blocking probability is a very good way to determine how good an architecture is.

This testbed can either define any physical topology, use any wavelength conversion scheme or accept any routing scheme. We will use the testbed to evaluate the performance of different topologies provided by the user and study the effect of wavelength conversion and routing schemes. To study how useful our testbed is, we are proposing the following thesis statement:

Wavelength conversion is useful in all-optical networks.

1.2 Results Reported in the Thesis

In this thesis, we can establish, through simulation, that wavelength conversion is useful in wavelength reuse and improves the efficiency of all-optical networks. Limited wavelength conversion is as good as full wavelength conversion in terms of their performances. Limited wavelength conversion is relatively less expensive and is therefore a very attractive option.

1.3 The Organization of the Thesis

Chapter 2 reviews the background information about all-optical networks. Chapter 3 describes our problem specification and how we are going to approach the problem using illustrative examples. Chapter 4 focuses on our network simulation and gives instructions on using the testbed to simulate different architectures, routing schemes and wavelength conversion schemes. Chapter 5 shows the results of our simulation experiments. Chapter 6 presents some future work and our recommendation. Finally chapter 7 concludes our work with a critical summary. Appendix A gives details on how to run the testbed with an example to show the step-by-step instructions to follow.

Chapter 2 REVIEW OF LITERATURE

A network using optical components is capable of transferring data at much higher rates and more reliable than that of using conventional electronic counterparts. In this section we will describe the state-of the-art fiber optic technology including the description of all-optical networks, a multiplexing scheme called Wavelength Division Multiplexing (WDM) and different topologies used in computer communication.

2.1 What is an All-Optical Network?

2.1.1 An Optical Fiber Cable

The optical fiber is one of the preferred transmission medium for telecommunications. In optical fiber technology, information is converted from electrical signals into optical signals at the source. This information travels along the optical fiber cable in the form of frequency modulated light and is re-converted into electrical signals at the destination.

The structure of a fiber cable is given below [1]:

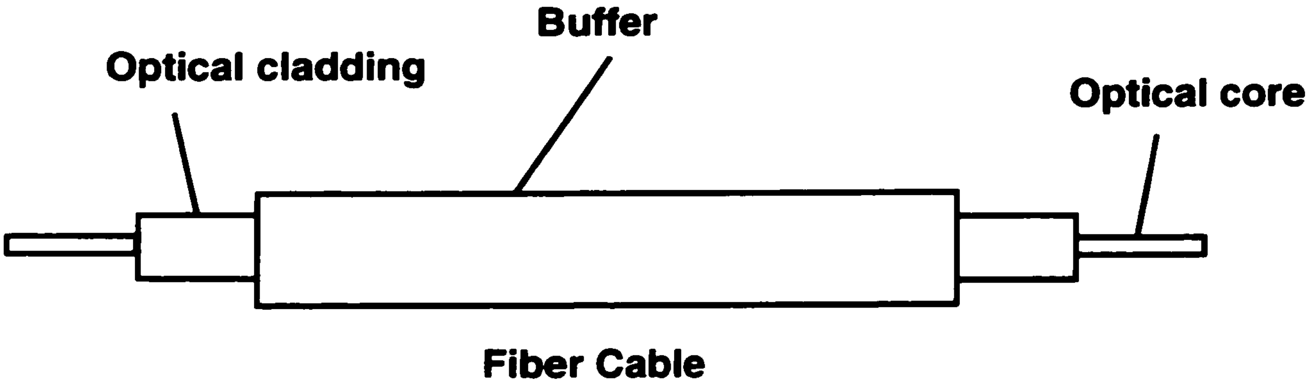


Figure 1 A fiber cable

An optical core is the innermost part of a fiber cable. An optical cladding is the middle layer, which is used to confine light to the core with a low refractive index. Finally the buffer serves as a protective layer to prevent the core and cladding from damage. Usually, a bunch of fibers can be packed in a cable.

To communicate from a source to a destination, a transmitter at the source generates optical signal. This signal is injected by the transmitter into the fiber optic core and travels along the cable by reflection. It hits the core-to-cladding interfaces at a critical angle and results in total internal reflection. Light power is thus conserved. Rays of light are gathered into modes, which become possible paths for a light ray travelling down the fiber. An optical fiber usually can allow as few as only one mode and as many as tens of thousand of modes. The number of modes create a significant factor such that more modes usually generate dispersion. Dispersion results in the spreading out of light power through the optical fiber[1].

Structures of fiber can be classified into three different categories. They are multimode stepped index fiber, multimode grade index fiber, and single-mode fiber [1][5]. They will be explained in the following paragraphs.

The multimode stepped index fiber is the simplest optical fiber such that rays travel along the core with glass of a lower refractive index. It allows multiple beams to occur simultaneously. However, this results in a high dispersion and low bandwidth of lightwave transmission.

The multimode grade index fiber is a varied implementation of multimode stepped index fiber. Rays are continuously refocused along the fiber because it uses a variable refractive index across the cross-section of the glass. This comes up with a lower dispersion and higher bandwidth than multimode stepped index fiber.

In order to achieve the highest bandwidth and lowest attenuation, the single-mode fiber is what we choose to be the preferred medium to transmit lightwave signals. It consists of a small core only to allow a single mode of light propagation. Such fiber can have an attenuation as low as 0.16 db/km [21]. However, this type of fiber is the most expensive one among them. Currently the most favorable fiber optic element for networking use is the 62.5/125 multimode grade index fiber, with the core diameter being 62.5 micrometers and the cladding being 125 micrometers.

Advantages of fiber optic technology are as follows [10]:

1. huge bandwidth (nearly 50 terabits per second (Tbps)),
2. low signal attenuation (as low as 0.2 dB/km),
3. low signal distortion,
4. low power requirement,
5. low error rates,
6. low material usage,
7. small space requirement, and
8. low cost.

The bandwidth of an optical network is superior to electronic networks by providing 50 Tbps in bandwidth in optical domain compared to a few gigabits per second (Gbps) in bandwidth in its electronic domain. An enormous bandwidth in optical fiber achieves a significant improvement over copper wire by providing more information simultaneously.

Attenuation in optical networks is only as low as 0.2 db/km. Attenuation is the loss of power when the signal travels along the transmission medium. Light loses its power because of reflection along the fiber. An optical amplifier is required if there is a long distance transmission. However, since the signal loss in fiber optics can be made very small, the number of amplifiers and repeaters needed can be reduced.

In addition to its enormous bandwidth and low attenuation, fiber optics also provides low error rates. Fiber optic systems, on average, operate at bit error rates (BERs) of less than 10^{-11}. Since information is carried along the cable in pulses of light, the signal is immune to noise and electric currents. Information is not easily disturbed and hard to be tapped into. The fiber optic cable is proved to be quiet and secure. The small size and thickness of the fiber requires less physical space to be occupied compared to copper. Finally, fiber is made from sand, one of the cheapest and most readily available substances on earth. Therefore fiber is at low cost and needs little space to install.

2.1.2 An All-Optical Network

Networks can be categorized into three generations. The first generation networks use copper wire and microwave technologies. No optical devices are used because they are still in the experimental stage during the first generation. The second generation starts involving fibers as a transmission medium, but mainly maintains traditional electronic switching. As a result, bandwidth is not good enough for demanding networks such as graphical performance and supercomputer interconnections. Finally, third generation comes to solve the bottleneck of network traffic as it uses "all-optical" communication without switching from optical domain to electronic domain. In other words, communication between nodes exclusively in the optical domain is only possible in the third generation of lightwave networks. All-optical networks are superior compared to the previous generations since they allow nearly four orders of magnitude capacity increase with respect to the speed limitation of electronics.

The fundamental mechanism of all-optical communication network is a lightpath. A lightpath is a communication channel between two nodes in the all-optical network, and it may extend over more than one fiber link.

An example of using an all-optical network is the medical imaging in hospitals [11]. Since medical imaging asks for the transmission of uncompressed images due to the fact that it does not trust image-compression techniques, it becomes a bandwidth-intensive application. In addition, there is more graphic-oriented software available nowadays and many new applications involve multimedia features, such as audio and video. As a result, an all-optical network is suitable for high bandwidth communication.

2.2 Components of an Optical Network

In order to explain different devices in an all-optical network, a diagram of a ring network is given below and different components are described as follows:

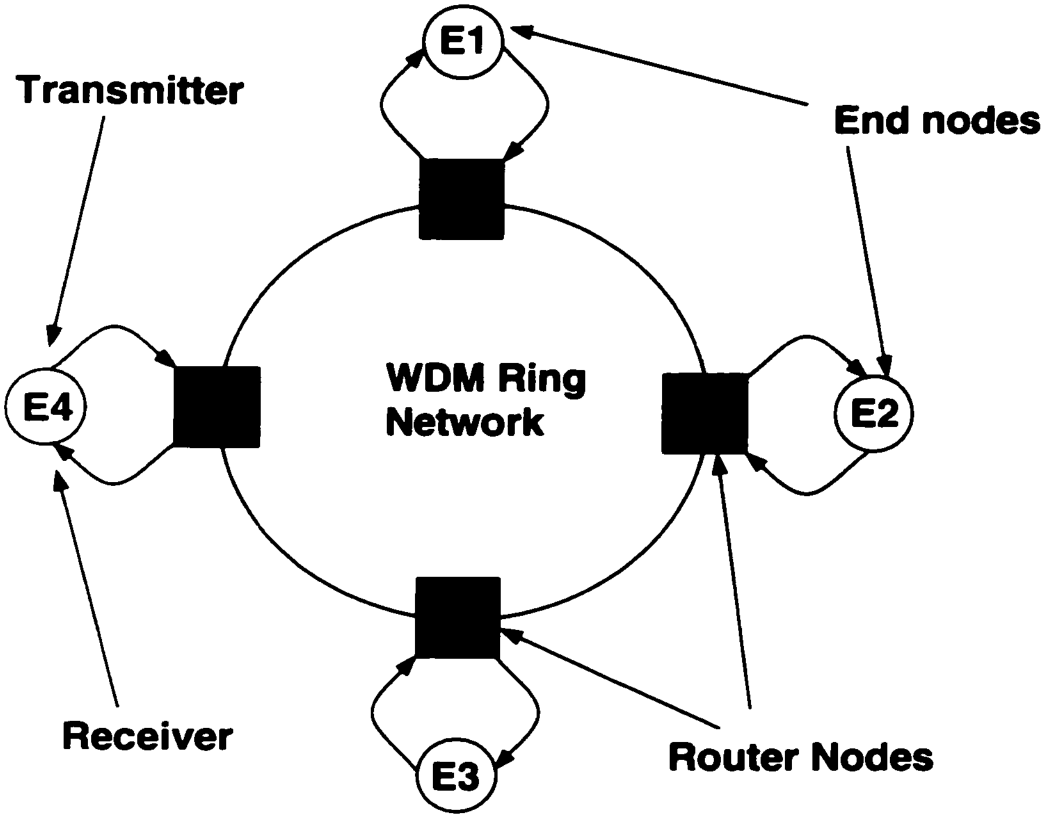


Figure 2 A ring network

End Nodes

An end node represents a source or a destination node in any communication. A computer is usually an end node in a small network. However, it may apply to a network in a huge Wide Area Network (WAN). A lightpath starts from one end node and finishes at another end node. In a typical optical network, each end node is connected to one or more router nodes in a network. An example of a ring network diagram is shown above.

Transmitters and Receivers

A transmitter is the device used to convert an electrical signal into light beams and send the light signal through a Light Emitting Diode (LED) or Laser Diode. A receiver changes incoming light beams into electrical signals read by end nodes. This conversion is generated by a phototransistor or photodiode. Transmitters and receivers can be either tunable or fixed frequency devices according to the type of application. Furthermore, they are usually grouped together and referred as transceivers.

Routers

A router is a switch with a number of fibers carrying input signals and a number of fibers carrying output signals. A router is used to determine the path used by a lightpath from a source to a destination. In other words, it sends an input signal to a particular output based on the router's switch setting. In the case of wavelength translation in a network, routers can change the carrier wavelength using a wavelength converter embedded inside the routers so that a communication can get through.

There are two types of routing strategies: static routing and dynamic routing. First we consider static routing. When static routing is employed, it uses a predefined path for every communication in the network. It is a relatively simple scheme because every communication has a fixed path to follow. However, this scheme takes a large number of wavelengths because all paths are set and all wavelengths are assumed to be used in designated routes.

Contrary to static routing, dynamic routing determines the path at run time. Therefore it is more flexible and economical use of wavelengths. However, it runs relatively slower than static routing because set-up time for dynamic routing is longer.

Wavelength Converter

As we mentioned earlier in routers, a wavelength converter can change a lightpath from one wavelength to another such that it can provide effective reuse of wavelength. In the diagram below, an example of a ring network embedded with wavelength converter is demonstrated.

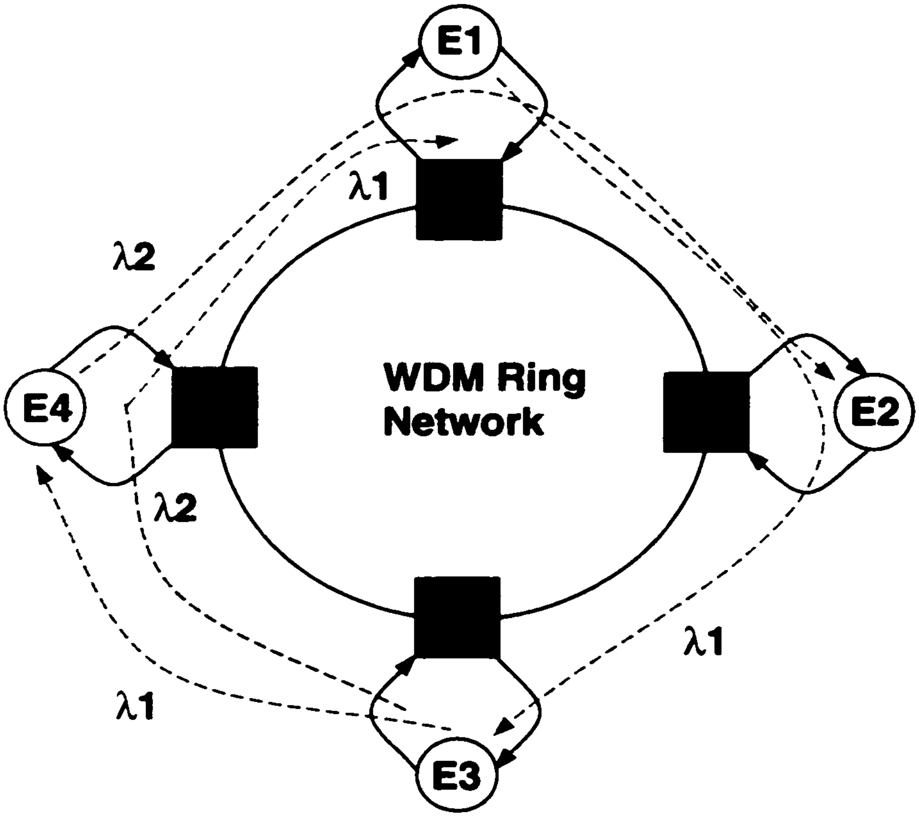


Figure 3 A wavelength converter in a ring network

In the ring network shown above, there are only two wavelengths available, namely λ_1 and λ_2. They are used according to the diagram as shown. As a result, there is no lightpath capable of travelling from node E3 to node E1 via node E4 because nodes E4 and E2 have used up both wavelengths. However, if a router in E4 is equipped with a wavelength converter, wavelength λ_2 can be used for a lightpath from E3 to E4 and another wavelength, λ_1, can thus be utilized and passed from E4 to E1. Such mechanism helps reduce cost as an additional wavelength will increase the cost of a network.

2.3 Wavelength Division Multiplexing (WDM)

In order to utilize the huge bandwidth of optical communication, we propose different access techniques to introduce concurrrency among users in the network. There are three major categories to achieve this concurrency according to either wavelength [Wavelength Division Multiplexing (WDM)], time slots [Time Division Multiplexing (TDM)], and wave shape [Code Division Multiplexing (CDM)] [12]. Among the three techniques, WDM is the most feasible implementation in optical communication network in terms of technology available today [10]. More importantly, it is easy to set up and has a simple design consideration.

The theory behind WDM is that the optical bandwidth of a fiber is packed into a bundle of smaller-capacity channels and, by tuning its transceivers, signals are transmitted into or received from those channels. In other words, each wavelength in WDM supports a single communication channel. By allowing multiple WDM channels to coexist on a single fiber, a huge bandwidth can be achieved with appropriate network architectures, protocols, and algorithms.

In a broadcast-and-select network, multicasting is an important concept. Multicasting is defined such that when a source transmits information using a particular wavelength λ_1, more than one receiver in the network can be tuned into wavelength λ_1 and are able to pick up the information stream.

There are cases where WDM is employed for a multiuser communication in a multiple access environment. As a result, Wavelength Division Multiple Access (WDMA) network is defined as a local optical network that utilizes WDM[10].

Research and development on the optical WDM networks has become a major topic over the past few years. Recent publications[1] have reviewed this popular topic in much detail. There were also WDM workshops held during recent conferences such as the Optical Fiber Communication (OFC '97) conference and IEEE International Conference on Communications (ICC '96). All research and workshops reveal that WDM will be the chosen network in the next decade.

A significant proof of cost reduction using WDM is found in [10] so that it applies to three metropolitan-area networks belonging to various telephone companies. The cost savings of using WDM compared to not using WDM in the

[1] JLT96 = IEEE/OSA Journal of Lightwave Technology, Special issue on Multiwavelength Optical Networks, Vol. 14, No. 6, June 1996

JSAC96 = IEEE Journal on Selected Areas in Communications, Special issue on Optical Networks, Vol. 14, No. 5, June 1996

JLT93 = IEEE/OSA Journal of Lightwave Technology, Special issue on Broadband Optical Networks, Vol. 11, No. 5/6, May/June 1993

JSAC90 = IEEE Journal on Selected Areas in Communications, Special issue on Dense Wavelength Division Multiplexing Techniques for High Capacity and Multiple Access Communication Systems, Vol. 8, No. 6, Aug. 1990

JHSN95 = Journal of High-Speed Networks, Special issue on Optical Networks, Vol. 3, Nos. 1 — 2, January — April 1995 networks are more than 16% with the actual “dollar values” worth $86, 000, 000. Therefore, WDM will be employed in every major project in telecommunication in the near future.

WDM networks are classified according to two categories: broadcast-and-select network and wavelength routing network. Traditionally broadcast-and-select networks focus on local networking applications and wavelength routing networks are meant for wide-area deployment [14]. Both will be explained in subsequent sections.

2.4 Broadcast-and-Select Network

A broadcast-and-select network consists of two types of network. The first network is called a single-hop system, whereas the second one is called a multihop system. The choice of using either system depends on different requirements and implementations of the network [21].

2.4.1 Single-Hop Systems

A single-hop system is defined to pass messages from a source to a destination in one hop. In other words, it means a direct communication from a source to a destination using only one lightpath. Therefore, a single-hop system is also known as an all-optical network [14] because its transmission is in an all optical domain. It results in a very fast communication. However, since there is only one hop in every connection, the routing procedure is done by tuning transceivers such that it connects the source and destination without intervention by immediate nodes. This requires a fast tuning time for transceivers and a substantial amount of dynamic coordination between nodes. Transmitters from the source node and receivers from the destination node must be tuned to the same wavelength in order for a successful communication.

An example of a broadcast-and-select WDM network is shown in Fig. 4 below.

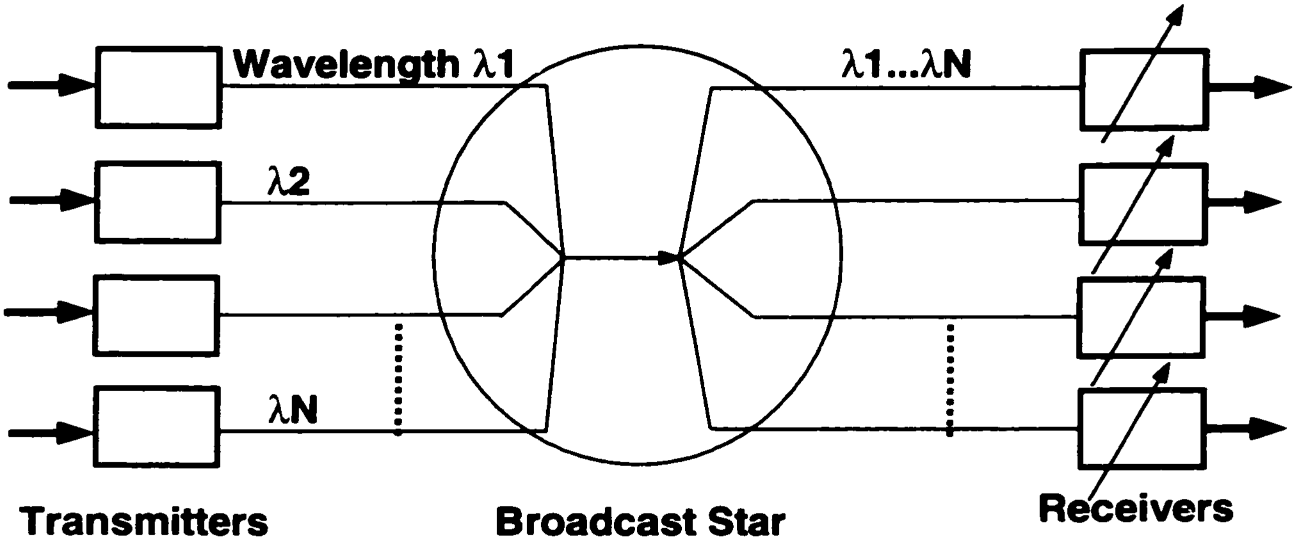


Figure 4 A single-hop system

Tunable transmitters are defined as the range of wavelength for transmission being tunable, whereas fixed transmitters are fixed for the range of wavelength for transmission. Similarly, tunable and fixed receivers are defined respectively.

Whether to choose a tunable or fixed transceiver relies on a trade-off between the tuning range of transceivers and their tuning times. The tuning time for tunable transceivers definitely slows down the process of transmission, but it provides flexibility to the system.

In order to classify different single-hop systems, they are based on two categories, those which use pretransmission coordination and those which do not. Pretransmission coordination systems specify a single and shared control channel to deal with the transmission requirements and data transfers through a number of data channels.

The key consideration for single-hop systems is that a protocol mechanism must be efficiently coordinated for data transmissions.

Single-hop systems are also based on whether the nodal transceivers are tunable or not. There are four possibilities [12]:

1. Fixed Transmitter(s) and Fixed Receiver(s) — (FT - FR)
2. Tunable Transmitter(s) and Fixed Receiver(s) — (TT - FR)
3. Fixed Transmitter(s) and Tunable Receiver(s) — (FT - TR)
4. Tunable Transmitter(s) and Tunable Receiver(s) — (TT - TR)

In the case of FT - FR structure, it is more suitable for multihop systems. For FT - FR and TT - FR systems, they may not require any pretransmission coordination because they have fixed receivers. Systems based on TT - TR structures are the most flexible, but the overhead cost should also be taken into consideration. Sometimes CC is labelled ahead of these classification such as CC - FT - FR. CC here stands for a control channel (CC)-based system where a control channel is found when there is a pretransmission coordination for the system.

Examples of single-hop systems under experimental WDM systems include LAMBDANET, RAINBOW and fiber-optic crossconnect. They also do not require pretransmission coordination. Protocols with no pretransmission coordination specify fixed assignment, partial fixed assignment protocols, random access protocols, and the PAC optical network. On the other hand, protocols with pretransmission coordination include extended slotted ALOHA and reservation ALOHA protocols, receiver collision avoidance (RCA) protocol and dynamic time-wavelength division multiple access protocol. The protocols mentioned above differ from one another based on how they are architecturally built and if their transceivers are tunable or not.

Further investigations on single-hop systems is given in [10][12].

2.4.2 Multihop Systems

Contrary to a single-hop system, a multihop system specifies a communication from a source node to a destination node involving a number of lightpaths normally with different wavelengths. Usually the path reaches a number of intermediate nodes before it routes to the destination node. Therefore buffering is needed at intermediate nodes. However, this buffering may drastically affect the speed of transmission. A good routing scheme focuses on the problem of determining the path from a source to a destination involving optimal numbers of immediate nodes in between.

Compared to single-hop systems, tuning time for transceivers is not an important factor for multihop systems. However, there are two other issues to be considered for a good multihop system. First, the logical topology must be optimal because it affects the routing algorithm. Second, the complexity for nodal processing must be small because it needs to travel a certain number of intermediate nodes before it reaches the destination node [11].

Multihop structures can be either regular or irregular. A regular topology maintains a regular connectivity pattern in a network. The topology can be derived either by a function or a predefined pattern. Examples include ShuffleNet, De Brujn Graph, and Mahattan Street Network. Among them, de Bruijn Graph and Manhattan Street Network (MSN) will be explained in following sections.

Another type of topology is an irregular topology. As its name suggests, an irregular topology does not have a concrete structure or connectivity pattern to explain how nodes are connected in a network. Therefore, it requires a complicated routing algorithm because of its irregular pattern.

Furthermore, physical and logical topology of multihop systems are important in the multihop systems. A physical topology and a logical topology of a network symbolize two different representations of the network as explained below.

A physical topology is defined as an actual connection in the multihop system. Usually a node is connected to other components before it reaches its destination, such as routers and other intermediate nodes. Therefore the physical topology represents the actual implementation of the network.

On the other hand, a logical topology describes how information is passed from one node to another node and does not consider how the system is set up. Logical topology is also known as virtual topology. It eliminates other components in the system. In other words, it represents a straightforward communication between source and destination.

Examples of physical and logical topology are shown in Fig. 5 and 6.

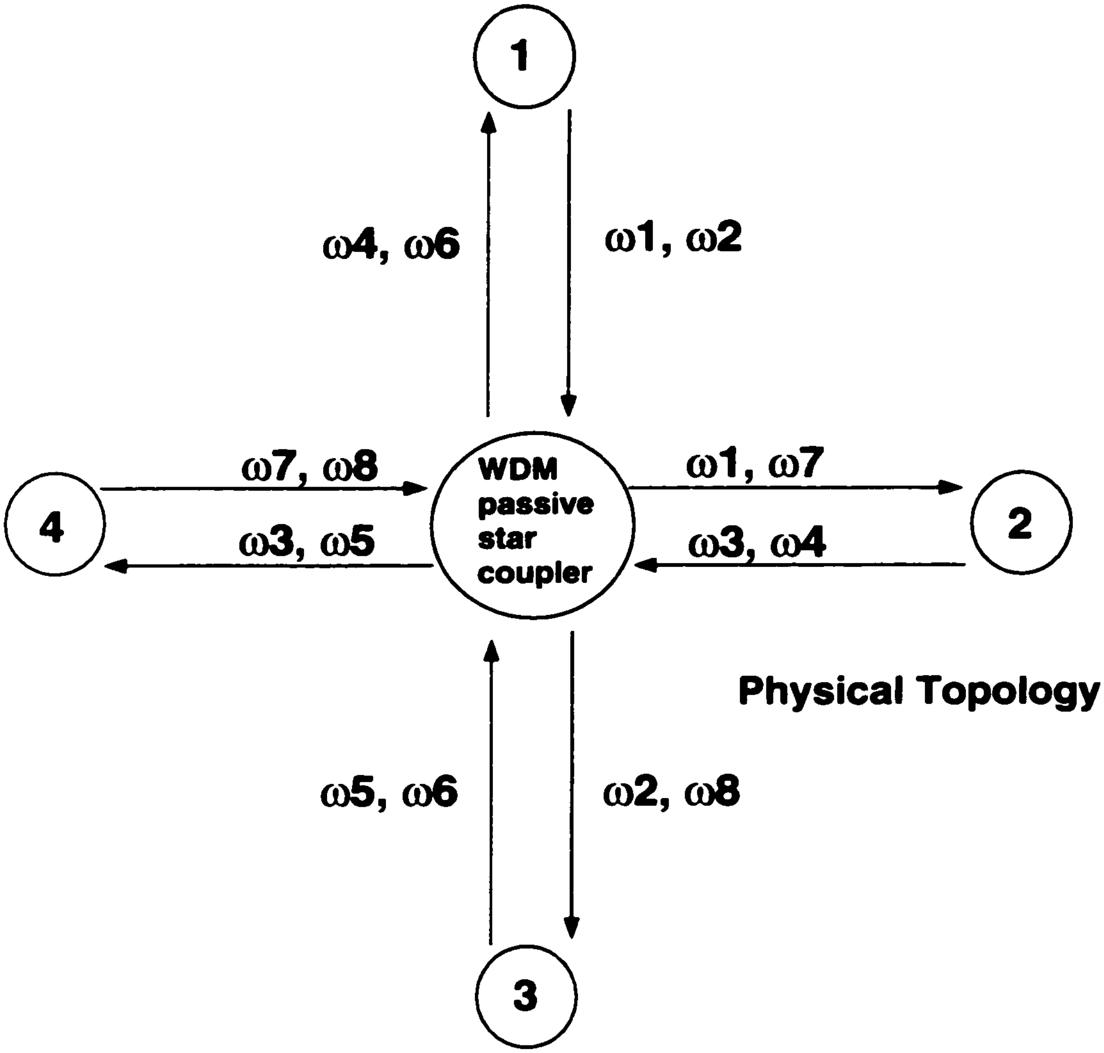


Figure 5 A physical topology for a multihop system

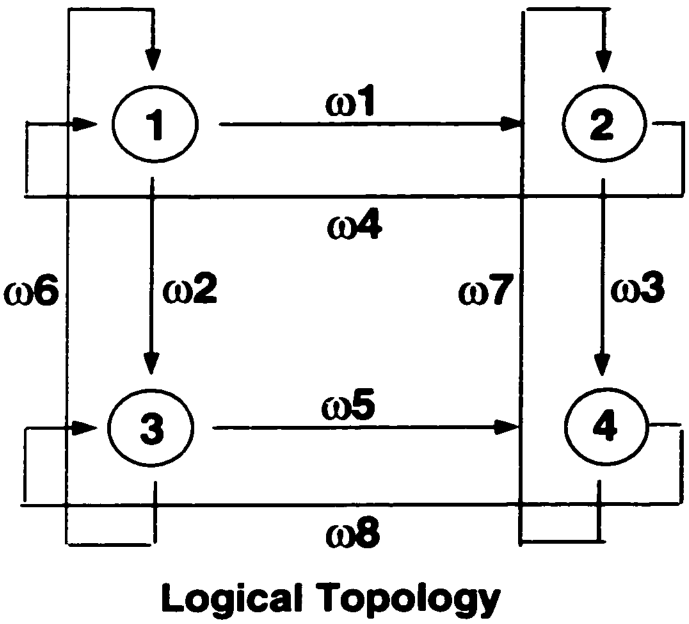


Figure 6 A logical topology for a multihop system

Further research on multihop systems can be found in [10][11].

2.5 Wavelength Routed Network

A wavelength routed network basically consists of reconfigurable wavelength sensitive devices interconnected by WDM fiber links. Their purpose is to route optical signals in the network. Examples of such devices are grating multiplexers and demultiplexers. Each link carries information on several carrier wavelengths. The routing nodes usually have end-nodes attached that form the sources and the destinations respectively for the network traffic. Each wavelength can be switched and routed independently from the others at each routing node. A lightpath is chosen on a particular wavelength and path through the network. Two lightpaths that share a common fiber link must use different wavelengths for maintaining wavelength continuity in the network unless wavelength conversion is established. The routing node must be able to route a signal coming in on one wavelength at an input port to any other output port, independent of signals at other wavelengths. This routing can be either static or dynamic, as explained in section 2.2. Meanwhile, amplifiers may still be used to overcome excessive losses in the routing nodes and the fiber-link losses.

An example of a wavelength routed network is given in the Fig. 7. There are four communications established in the network. Wavelength λ_1 is used from node 1 to node 2 while it is also being used from node 2 to node 5. It does not cause any conflicts because each one of them occupies distinct fibers even though they are using the same wavelength simultaneously. This enables wavelengths to be reused in the network. Wavelengths can be reused as long as there is no wavelength used on the same link simultaneously

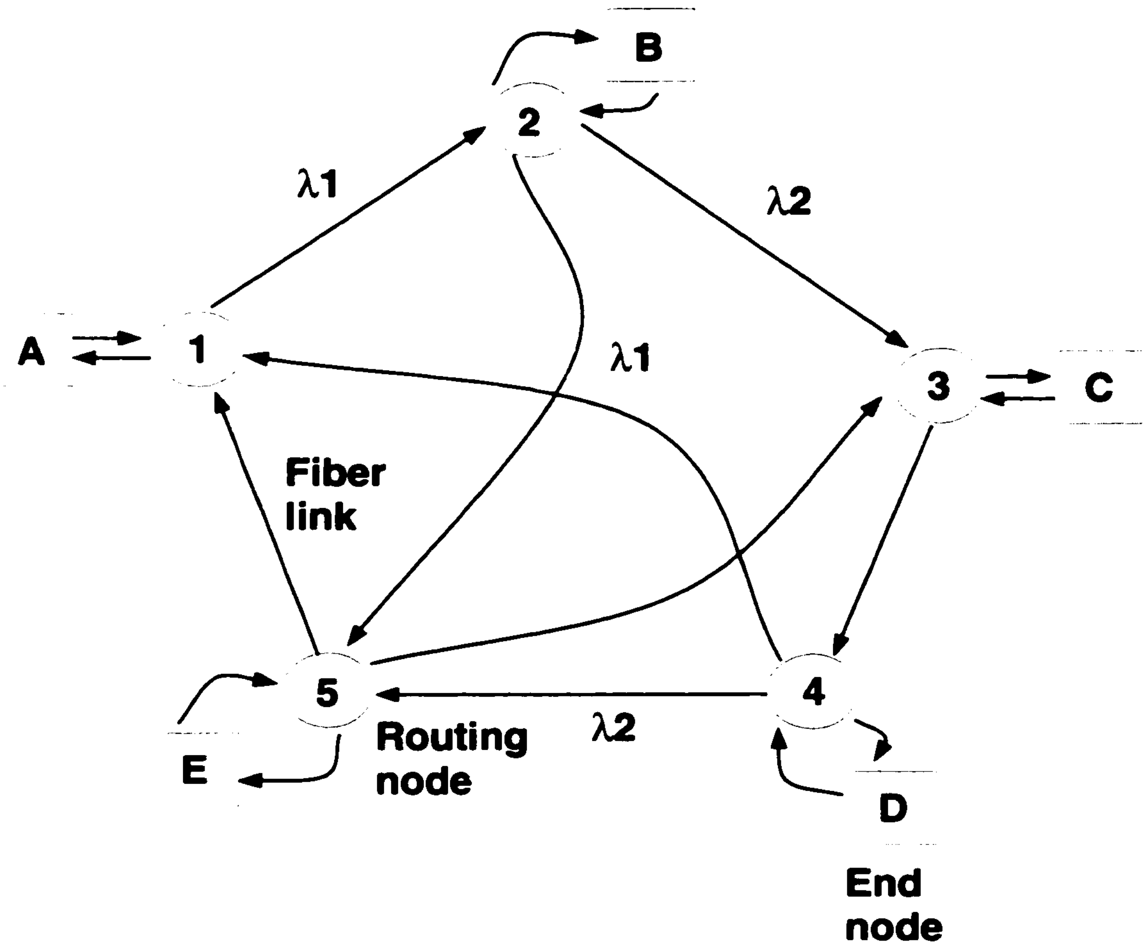


Figure 7 A wavelength routed network

In order to develop good routing schemes, interesting research on routing and wavelength assignment (RWA) have started in different research groups [10][18]. Designing good logical topologies is also an important aspect of studying wavelength routed networks [6][17][26].

Wavelength routed networks are meant to be used in wide area networks (WANs). Its support of packet switching, reasonable hardware cost and simplicity of network control makes the network competitive [14].

A wavelength routed network avoids dispersion. Unlike a broadcast-and-select network, where signals are sent to every station in the network, there is only one destination for a wavelength routing network. It is cheaper since it involves less transmission power.

Further discussion on wavelength routed network can be found in [4][8][15][17][14][16].

2.6 Wavelength Conversion

To begin the study of wavelength conversion technologies, it is good to start with IEEE/OSA Journal of Lightwave Technology Vol. 14, No. 6 June 1996, which is a special issue on Multiwavelength Optical Technology and Networks. It has several extensive overview papers on wavelength conversion.

2.6.1 Wavelength Conversion Technologies

The concept of wavelength conversion in all-optical networks is to reduce the load of a heavy network by switching from one wavelength to another whenever needed. In a large network, there is not enough wavelength for every number of nodes. Thus, the blocking probability increases due to possible wavelength contention when two lightpaths at the same wavelength are to be routed at the same output. One possible solution to overcome this limitation is to convert signals from one wavelength to another. This is how the concept of wavelength conversion rises up. Wavelength conversion raises the full potential of the wavelength dimension in WDM networks [24]. This can be called a wavelength-convertible network [7].

An example of how wavelength conversion is used is illustrated below in Fig. 8:

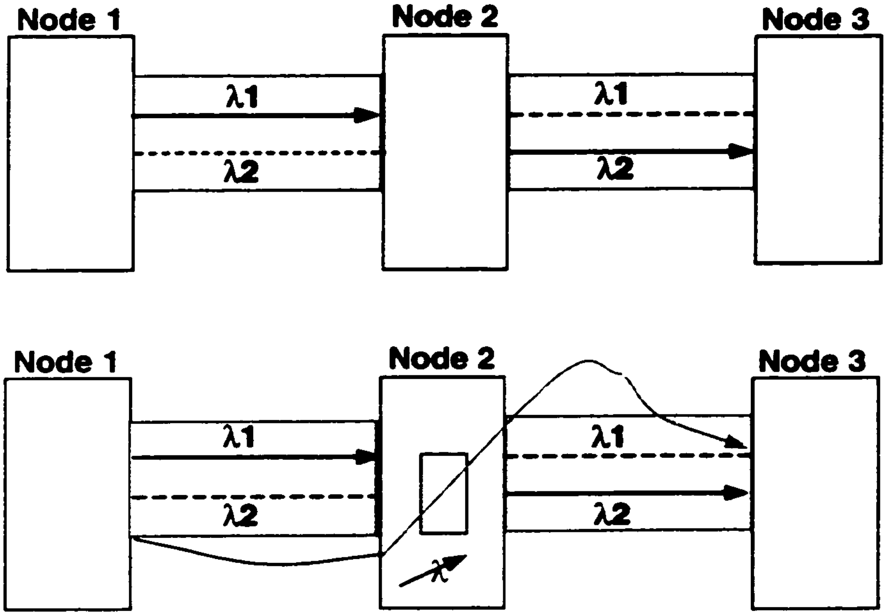


Figure 8 Wavelength Conversion

A wavelength-continuity constraint is defined as using the same wavelength all the way from a source node to a destination node. In the case of an absence of any wavelength conversion devices, a lightpath is required to maintain the same wavelength channel throughout its path in the network in order to reserve the wavelength-continuity constraint property of the lightpath. However, this constraint may not be necessary if wavelength conversion is allowed in the network.

A blocking probability symbolizes how successful a network is by noting how many calls are being blocked out of all attempted communications. Sometimes a call is blocked because all WDM channels are used up from a source node to a destination node. As a result, a good network should always look for a low blocking probability. In other words, the lower the blocking probability, the better the network.

Wavelength conversion can be classified into two different categories. A full wavelength conversion is capable of converting a wavelength from all sets of existing wavelengths in the network into another wavelength, whereas a limited wavelength conversion just allows such a conversion chosen from a subset of existing wavelengths in the network. They will be explained in detail later.

There are three types of wavelength converters available based on mapping function and the form of control signals. They are classified as optoelectronic, optical gating, and wave-mixing respectively.

Firstly, optoelectronic (O/E — E/O) wavelength conversion is the most straightforward and mature method of all wavelength conversion techniques. This, however, is slow in terms of its wavelength conversion operation.

Secondly, optical grating wavelength conversion uses a semiconductor optical device which converts its characteristics according to the intensity of the input signal.

Thirdly, wave-mixing wavelength conversion is least explored but provides strict transparency because it keeps both phase and amplitude information. This is the result from a nonlinear optical medium when there is more than one wave present. This wave-mixing effect generates another wave that comes from the product of the interacting waves.

2.6.2 Characteristics of Wavelength Conversion

Good wavelength converters should have the following characteristics [2][9]:

1. Transparency to the bit rate and signal format.
2. An unchirped output signal with both a high extinction ratio and a large signal-to-noise ratio.
3. Conversion for both shorter and longer wavelengths.
4. Simple implementation.
5. Insensitivity to input signal polarization.
6. Fast generation time of output wavelength.
7. Optimum input power levels.
8. Possibility of no conversion.

Advantages of wavelength converters[2][3]:

1. Wavelength reuse.
2. Simplicity of channel reservation and resource allocation.
3. Signal regeneration and reshaping can be done through wavelength converters instead of amplifiers.
4. Enhancement of flexibility of the network.
5. Optimize routing functions together with optical filters.

Disadvantages of wavelength converters[22]:

1. Expensive devices.
2. Not yet commercially feasible.

2.7 Properties of Networks Tested in This Simulation

2.7.1 Ring Network

A ring network is a network with nodes connected to each other in a circle. Here a 4–node ring network is shown in Fig. 9.

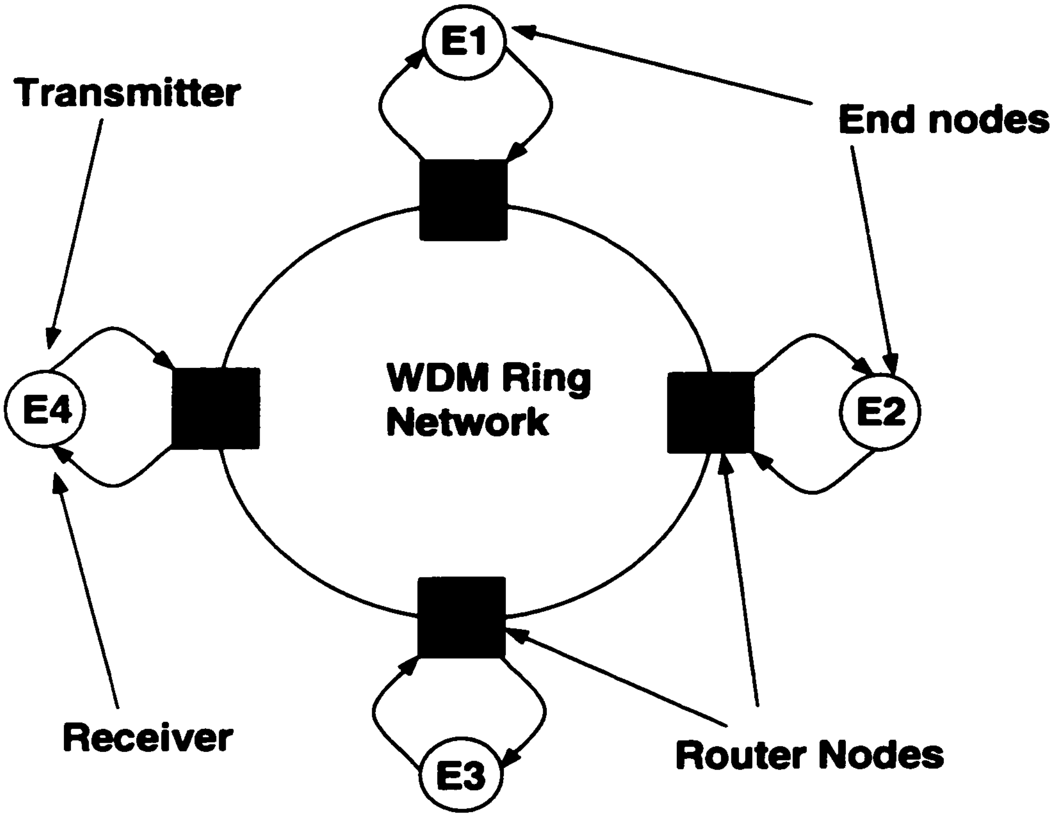


Figure 9 A ring network

The ring network is doubly linked so that every node is connected to its previous node and its successor node. Its bidirectional structure allows only one alternate path for any source-destination pairs. The shortest path depends on the shortest distance either in the forward direction or in the backward direction.

Example

In a ring network with 10 nodes, for a source node 7 and a destination node 3, the path is either

7 → 8 → 9 → 0 → 1 → 2 → 3 so that it goes forward

or

7 → 6 → 5 → 4 → 3 so that it goes backward.

Since the backward direction is shorter than the forward direction in distance, the backward direction is chosen as the path to be used.

2.7.2 De Bruijn Graph

A de Bruijn graph G (Δ, D) is a directed graph of Δ^D nodes with a set of nodes with a degree of Δ and diameter D [20]. Any node S in G (Δ, D) de Bruijn graph may be defined as Δ—ary representation such as an vector of D digits $d_D \ldots d_2\ d_1$ where each digit is between 0 and $\Delta - 1$. For example, a (2, 3) graph thus consists of $\Delta^D = 2^3 = 8$ nodes and its binary representation is as follows:

Node number	Binary representation	Derivation of the node from binary representation into node number
0	000	$2^2 * 0 + 2^1 * 0 + 2^0 * 0 = 0$
1	001	$2^2 * 0 + 2^1 * 0 + 2^0 * 1 = 1$
2	010	$2^2 * 0 + 2^1 * 1 + 2^0 * 0 = 2$
3	011	$2^2 * 0 + 2^1 * 1 + 2^0 * 1 = 3$
4	100	$2^2 * 1 + 2^1 * 0 + 2^0 * 0 = 4$

Table 1 Binary representation of (2, 3) De Bruijn Graph (Continued) . . .

5	101	$2^2 * 1 + 2^1 * 0 + 2^0 * 1 = 5$
6	110	$2^2 * 1 + 2^1 * 1 + 2^0 * 0 = 6$
7	111	$2^2 * 1 + 2^1 * 1 + 2^0 * 1 = 7$

Table 1 Binary representation of (2, 3) De Bruijn Graph

Fig. 10 illustrates the connectivity pattern of a G(2, 3) de Bruijn graph.

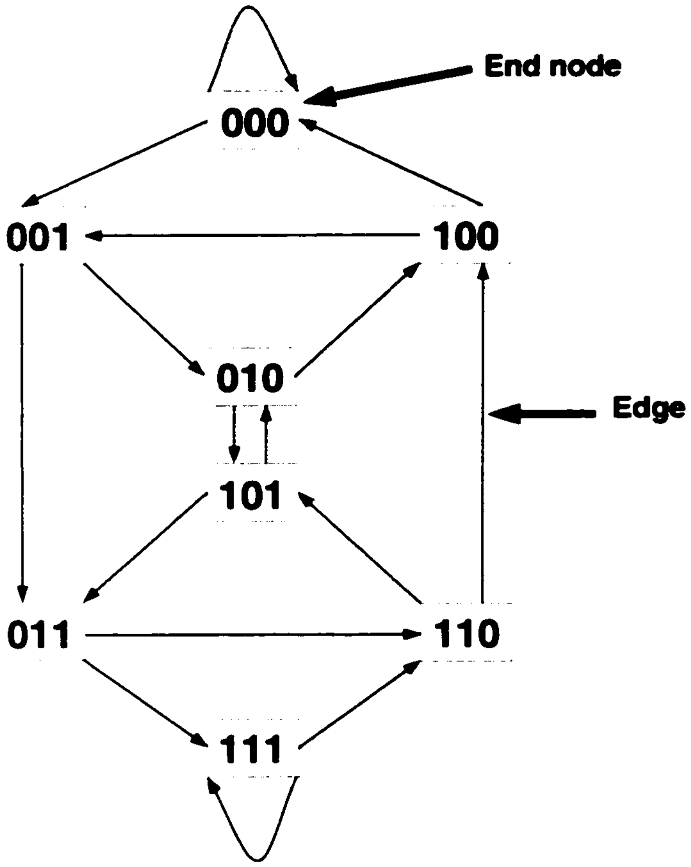


Figure 10 A (2, 3) De Bruijn Graph

The number of nodes can be determined by:

$N = \Delta^D$ where $\Delta \geq 2$ and $D \geq 2$

The set of nodes $\{0, 1, 2, ..., \Delta - 1\}$ with an edge from node $(a_1, a_2, ..., a_D)$ to node $(b_1, b_2, ..., b_D)$ if and only if the following condition satisfies:

$b_i = a_{i+1}$ for $1 \leq i \leq D - 1$ and a_i, b_i belongs $\{0, 1, 2, .., \Delta - 1\}$

In other words, there is an edge from a node A to a node B if the suffix of length $D - 1$ of the vector representing A matches the prefix of length $D - 1$ of the vector representing B.

In [20], a formal definition for the procedure to find the shortest path between any source node $S = s_1\ s_2\ ...\ s_D$ and any destination node $D = d_1\ d_2\ ...\ d_D$ is given as follows:

1. Find the smallest value k such that $(d_1\ d_2\ ...\ d_{D-k}) = (s_{k+1}\ s_{k+2}\ ...\ s_D)$
2. The shortest path between S and D is determined by the following pattern:

   $s_1\ s_2\ ...\ s_D \rightarrow s_2\ s_3\ ...\ s_D\ d_{D-k+1}$

   $s_2\ s_3\ ...\ s_D\ d_{D-k+1} \rightarrow s_3\ s_4\ ...\ s_D\ d_{D-k+1}\ d_{D-k+2}$

   ...

   $s_k\ s_{k+1}\ s_{k+2}\ ...\ s_D\ d_{D-k+1}\ d_{D-k+2}\ ...\ b_{D-1} \rightarrow s_{k+1}\ s_{k+2}\ ...\ s_D\ d_{D-k+1}\ d_{D-k+2}\ ...\ b_D = d_1\ d_2\ ...\ d_{D-k+1}\ d_{D-k+2}\ ...\ d_D$

Example:

In this example, we use a (3, 4) De Bruijn Graph to describe the state transition from source to destination. The graph thus consists of 81 nodes (i.e. $3^4 = 81$).

Here we have a source-destination pair from node 2012 to node 0222. As the diagram shows, we can see the state transition in every hop. A hop is a state change from one node to another. From node 2012 to node 0120, the last three digits from source node and the first three digits from successor node remains the same. This applies to all transitions from source to destination.

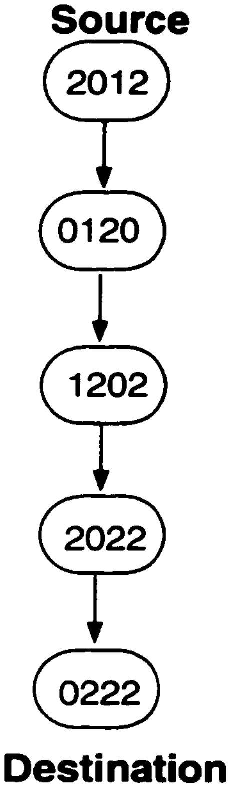


Figure 11 Node transition for (3, 4) De Bruijn Graph

Another routing scheme we have considered is alternative path routing [19]. The purpose of alternative routing is to provide several alternative paths rather than the shortest path in the establishment of the lightpath because there may be blocked communications in the shortest path. In general, for a (Δ, D) de Bruijn Graph, there are Δ paths of maximum length of D + 1. For any source node A = $a_1\ a_2\ \ldots\ a_D$ and any destination node B = $b_1\ b_2\ \ldots\ b_D$, there are Δ alternative paths from A to B listed below:

1. Path 1:

$a_1\ a_2\ \ldots\ a_D \rightarrow a_2\ \ldots\ a_D\ 0 \rightarrow \ldots$ shortest path $\ldots \rightarrow b_1\ b_2\ \ldots\ b_D$

2. Path 2:

$a_1\ a_2\ \ldots\ a_D \rightarrow a_2\ \ldots\ a_D\ 1 \rightarrow \ldots$ shortest path $\ldots \rightarrow b_1\ b_2\ \ldots\ b_D$

3. . . .

4. Path Δ:

$a_1\ a_2\ \ldots\ a_D \rightarrow a_2\ \ldots\ a_D\ (\Delta - 1) \rightarrow \ldots$ shortest path $\ldots \rightarrow b_1\ b_2\ \ldots\ b_D$

Example:

In a (3, 4) de Bruijn graph, there are three alternative paths between node 2012 and node 0222 as follows:

1. 2012 → 0120 → 1202 → 2022 → 0222
2. 2012 → 0121 → 1210 → 2102 → 1022 → 0222
3. 2012 → 0122 → 1220 → 2202 → 2022 → 0222

2.7.3 Manhattan Street Network

An M * N Manhattan Street Network consists of M rows and N columns with adjacent row links and column links connected to each other in two directions. In other words, it connects to the next row nodes and column nodes in both forward and backward direction to form a mesh structure.

An example of 4 * 4 Manhattan Street Network is given below in Fig. 12.

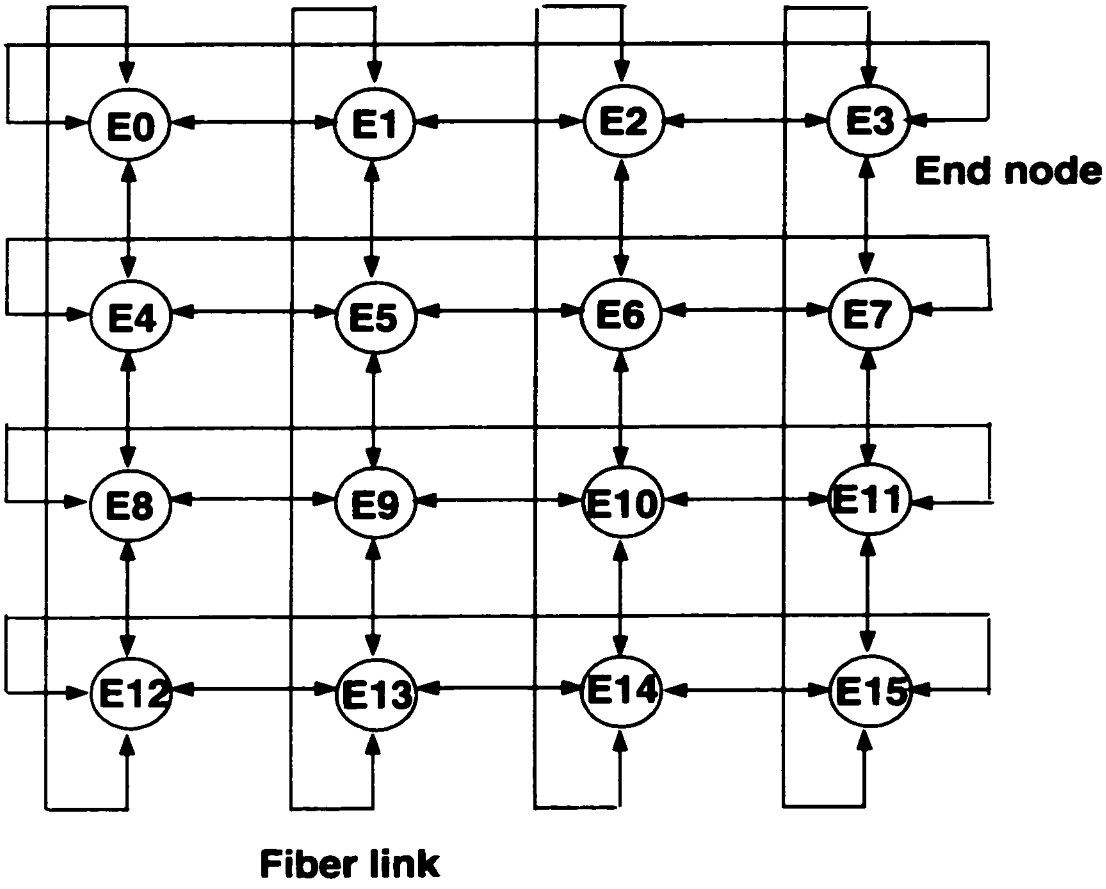


Figure 12 A 4 X 4 Manhattan Street Network

The routing scheme adopted here shows that we go by row transition and then by column transition. The source node reaches the row that the destination node is located, and moves to the column where the destination node is.

Example:

In the diagram shown above, we assume that E0 is the source and E14 is the destination,

In the row transition,

E0 → E4 → E8 → E12 involves three hops in forward direction, whereas

E0 → E12 involves only one hop to reach the destination row.

Therefore we choose backward direction to go to the destination row.

Next, we calculate the column transition:

E12 → E13 → E14 takes two hops, and

E12 → E15 → E14 also involves two hops in backward direction.

As a result, either direction we can choose to reach the destination node. Here we take the forward direction because it is first considered.

2.7.4 An Arbitrary Irregular Topology

As its name suggests, an arbitrary irregular topology has no definite structure and pattern. Every edge carries a weight to signify the distance to travel from one node to another one. An example of an arbitrary irregular topology is given below in Fig. 13.

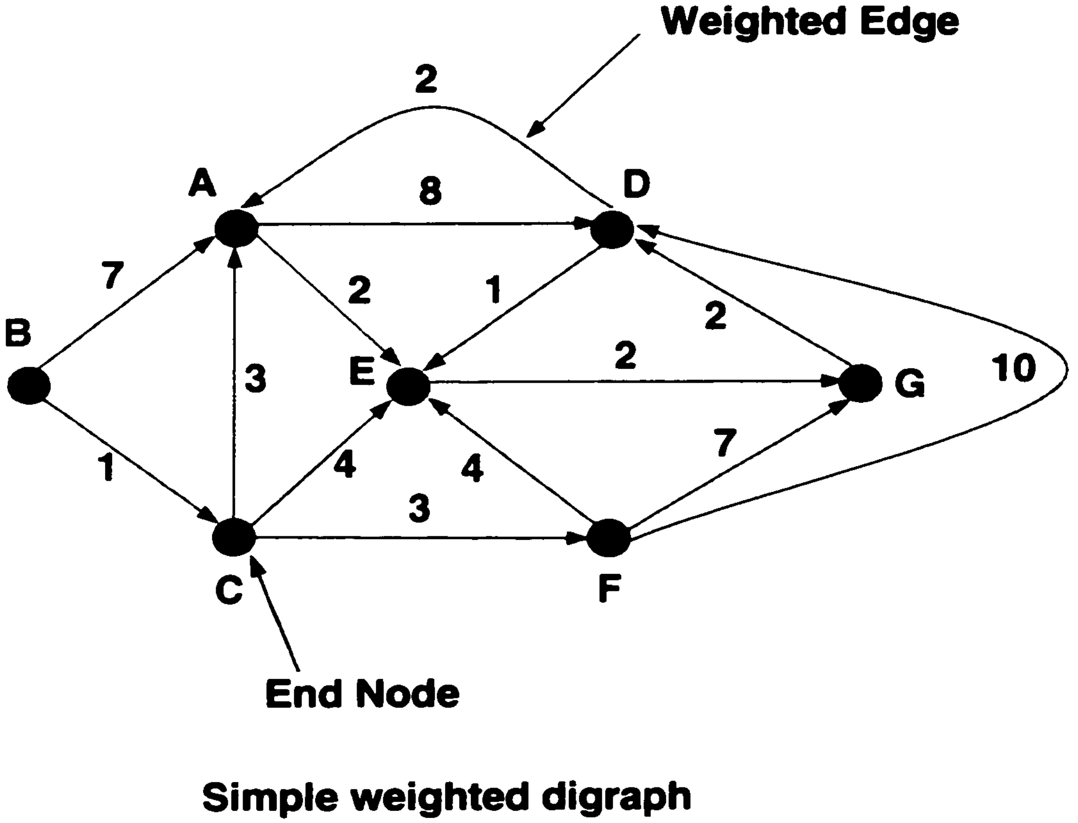


Figure 13 An arbitrary irregular topology

Certain communications cannot be established because there are no paths between sources and destinations. For example, from node A to node B, no communication can be established because there is no link from A to B at all. Furthermore, every edge has a designated direction where a doubly linked list is not available in this network. For example, there is an edge from node B to node C, but not vice versa because there is no edge from node C to node B in the network shown above.

We use Dijkstra's shortest-path algorithm for our routing scheme in this network [13]. Our purpose is to look for the shortest possible path and its length from a source node to a destination node. This algorithm labels all nodes of the network. At each stage in the algorithm some nodes get permanent labels and others temporary labels. When the algorithm starts, it assigns a permanent label 0 to the source node s, and a temporary label ∞ to the remaining $n - 1$ vertices. Next, another node has a permanent label in each iteration according to the rule as follows:

1. Every node j that is not yet permanently assigned a label gets a new temporary label whose value is derived by:

   Min [old label of j, (old label of i + d_{ij})],

   where i is the latest node permanently assigned a label, in the last iteration, and d_{ij} is the direct distance between nodes i and j. If i and j are not connected by an edge, then $d_{ij} = \infty$.
2. After all nodes have been evaluated, the smallest value among all the temporary labels will be chosen to become the permanent label of the corresponding node. If there is a tie, any one of the candidates for permanent labeling will be chosen.

Step 1 and 2 are repeated until the destination node d has a permanent label.

The first node to be permanently assigned is at a distance of zero from s. The second node to obtain a permanent label is the one closest to s. For the remaining n − 2 nodes, the next node to be assigned is the second closest node to s, and so on. In other words, the permanent label of each node is guaranteed the shortest distance of that node from s.

Next we are going to find the shortest path for the network. It can be simply constructed by working reversely from the destination node such that we go to that predecessor whose label differs exactly by the length of the joining edge until we reach the source node.

Example:

In the Fig. 13 shown above, we take B as the source node and G as the destination node.

After evaluation of the algorithm, we have the permanent label of each node[2]:

A = 4, B = 0, C = 1 D = 12 E = 5 F = 4 G = 7

Now we are looking for the shortest path from B to G. Starting from G, we have a permanent label 7 and we find E with a permanent label of 5 and an edge of 2. Therefore, E is the predecessor of G. And so on. Here we find the path from B to G:

[2] D is not a permanent label here because the distance of D from B is greater than that of G from B.

$${}^{0}B \xrightarrow{1} {}^{1}C \xrightarrow{4} {}^{5}E \xrightarrow{2} {}^{7}G$$

The number next to a node represents the permanent label of that node from the source node. The number up above the arrow stands for the weight for the edge connected from one node to another node. Therefore, ${}^{0}B \xrightarrow{1} {}^{1}C$ here means the permanent labels of B and C are 0 and 1, respectively, and the weight for the edge from B to C is 1.

Chapter 3 PROBLEM SPECIFICATION

3.1 Problem Definition

The objective of this thesis is to design an application that allows flexibility for users to insert different pieces into the testbed so that they can concentrate on their specification without going into too much detail to modify the testbed. In other words, it should be general enough to allow users to put their own specifications into the testbed to run for simulation. These specifications include the definition of physical topology, schemes for wavelength conversion and routing schemes used in the testbed. A block diagram of Fig. 14 visually explains our problem shown below.

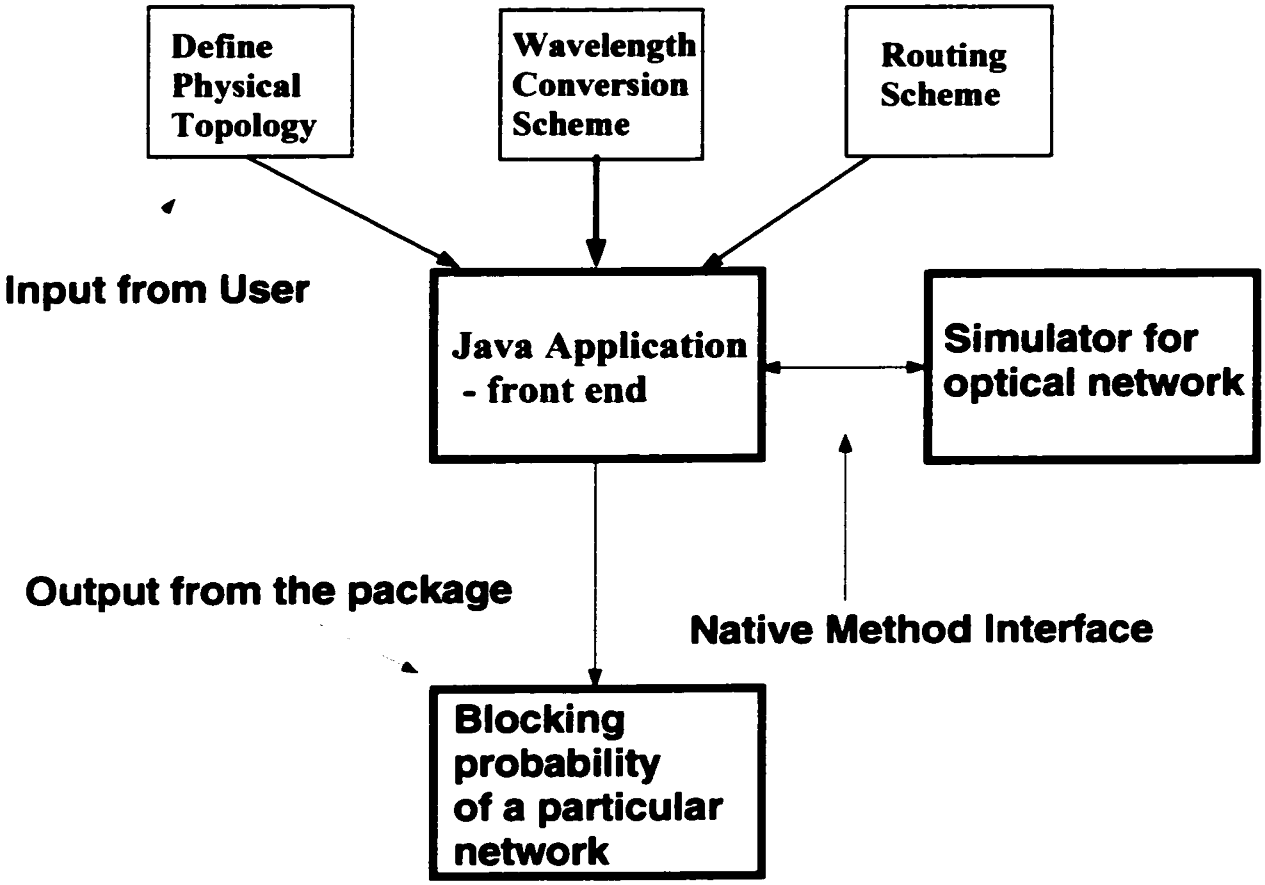

Define Physical Topology
Wavelength Conversion Scheme
Routing Scheme
Input from User
Java Application - front end
Simulator for optical network
Native Method Interface
Output from the package
Blocking probability of a particular network

Figure 14 Our objective in this thesis

We have constructed an application to accept users' specifications of their topologies and a simulation program to use all input data to generate the blocking probability of the particular network. When we test our topologies, one important consideration is that we have to keep the blocking probability low. The blocking probability relies critically on a number of factors. Our goal is to identify which factors are important to the network. Here are the factors we have considered:

1. Number of wavelengths in the network.
2. Number of wavelengths a transceiver in a node can be tuned to, (i.e., number of connection in a node).
3. Convertibility of a carrier wavelength (i.e. whether there is no, full or limited wavelength conversion).
4. Routing strategy.

In the following sections, we first list the steps to build the testbed. Next we will look at different physical topologies, then describe routing schemes in each topology. Finally we will explain different approaches for the wavelength conversion schemes we have used in our simulation with illustrative examples.

3.2 An Algorithm for the Testbed

Here is an algorithm for implementation of our testbed as follows[3]:

1. Accept the parameters from users in the user interface built in Java (a list of parameters will be given in next chapter "NETWORK SIMULATION").
2. Initialize network parameters such as "make" file and "define" file.
3. Send all data to a network simulator built in C.

[3] In this algorithm, we are only giving main steps here.

4. Run steps 5 to 8 in a certain number of times according to users' specification and keep track of the number of hits and misses from each trial.
5. Repeat steps 6 to 8 until certain conditions arises, such as no source with an available transmitter or no destination with an available receiver.
6. Randomly choose a source-destination pair with available transceivers.
7. Try to establish the path between the source-destination pair with various parameters such as wavelength conversion schemes and routing schemes.
8. If there is a successful connection between the pair, we increment the number of hits; otherwise we increment the number of misses.[4]
9. Write both the number of hits and misses back to an output file after the simulation finishes.
10. Calculate the blocking probability for each trial.

As mentioned in step 10, we are interested in the blocking probabilities of different topologies. Next section we will look at different topologies we use for the testbed, including their connectivity patterns, routing schemes and illustrative examples, respectively.

3.3 Input for Networks Tested in This Simulation

3.3.1 Ring Network

In our simulation, the characteristics of the ring network are as follows:

[4] For a successful connection, it will remain in the network until the network is "saturated" and another trial of simulation starts.

1. It has 300 nodes.
2. Range of number of wavelength to be used is from 300 to 650 with intervals of 50.
3. Number of connection is 5.
4. Maximum edge length is 300.
5. We have created an edge file for the ring network to specify the structure of the network.
6. There is a routing function for the ring network we have adopted as mentioned before.

3.3.2 De Bruijn Graph

In our simulation, the characteristics of the de Bruijn graph are as follows:

1. It has 1024 nodes, i.e. a (4,5) de Bruijn graph.
2. Range of number of wavelength to be used is from 20 to 30 with 1 as the interval.
3. Number of connection is 10.
4. Maximum edge length is 30.
5. We have established an edge function for the de Bruijn graph. The function is able to find the edge connecting any two nodes if there is a communication between the nodes.
6. A routing function for the de Bruijn graph is also created.

3.3.3 Manhattan Street Network

In our simulation, the characteristics of the Manhattan Street network are as follows:

1. It has 900 nodes.
2. Range of number of wavelengths to be used is from 100 to 240 with 20 as the intervals.
3. Number of connection is 20.
4. Maximum edge length is 80.
5. An edge function similar to the one generated for the de Bruijn graph has been generated.
6. A routing function especially made for the Manhattan Street network is created.

3.3.4 An Arbitrary Irregular Topology

In our simulation, we use a program to generate an irregular network such that it has the following properties:

1. It has 100 nodes.
2. Every node can have up to 3 edges connected to any arbitrary nodes.
3. No self loop is available.
4. Every edge has a weight value.

The characteristics of the irregular topology in our simulation are as follows:

1. Range of number of wavelengths to be used is from 100 to 600 with 50 as interval
2. Number of connection is 10
3. Maximum edge length is 100
4. Instead of using an edge function, we adopt an edge file generated from the program mentioned above. The file specifies a communication between any two nodes with a weight value.
5. We use Dijkstra's shortest-path algorithm for our routing scheme.

3.4 Our Approaches towards Different Wavelength Conversion Schemes

All the examples here explain our approaches towards different wavelength conversion schemes. Throughout this thesis, we will use the De Bruijn Graph for our illustration because we can keep at the same pace when we refer to different perspectives of examples.

Before we describe different wavelength conversion schemes, here are examples to signify the importance of wavelength conversion.

Example 1:

In this example, we use a De Bruijn graph G(2, 3) to represent an all-optical network. Each node is a computer in the network, whereas each edge stands for a communication link between two nodes. We assume that each edge can have a total of two wavelengths, λ_1 and λ_2, covered in a single optical fiber. There are two existing connections as shown in Fig. 15:

- a path from node 101 to node 111 using λ_1 through node 011
- a path from node 011 to node 110 using λ_2

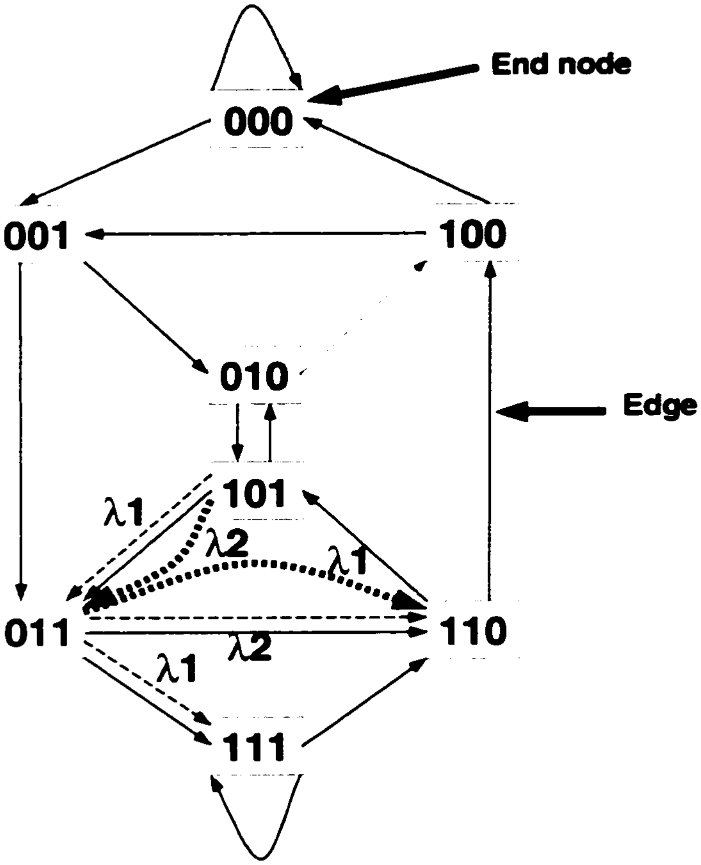


Figure 15 Example 1

Both paths use the shortest-path routing scheme to achieve the connections. Now we attempt to establish a connection such that it starts at node 101 and ends at node 110. If there is no wavelength conversion used in this network, such connection cannot be established successfully because there is no free wavelength that can be used along the path from the source to the destination. In other words, the shortest path for connection between 101 and 110 is 101 → 011 → 110, where λ_1 is already used on the edge 101 → 011 and λ_2 is already used on the edge 011 → 110. As a result, the connection is blocked.

However, if wavelength conversion is allowed, a connection is possible. First, λ_2 is employed on the edge 101 → 011. At 011, λ_2 will be converted to λ_1 by wavelength conversion. Afterwards, λ_1 is used on the edge 011 → 110. This results in a successful connection.

Example 2:

In this example, we look at a De Bruijn graph G(3, 4), a common example throughout our illustration. There are four existing wavelengths in the network. The source node is 2012, and the destination node is 0222. A selected path is shown in Fig. 16. We can demonstrate an edge 2012 → 0120, for instance, from the path where wavelength λ_0 is being used, but other wavelengths λ_1, λ_2, and λ_3 are available to be used on this edge. We assume that limited wavelength conversion is employed. In other words, each wavelength converter in every node can convert one wavelength to the next adjacent wavelength as shown in the conversion table in the diagram.

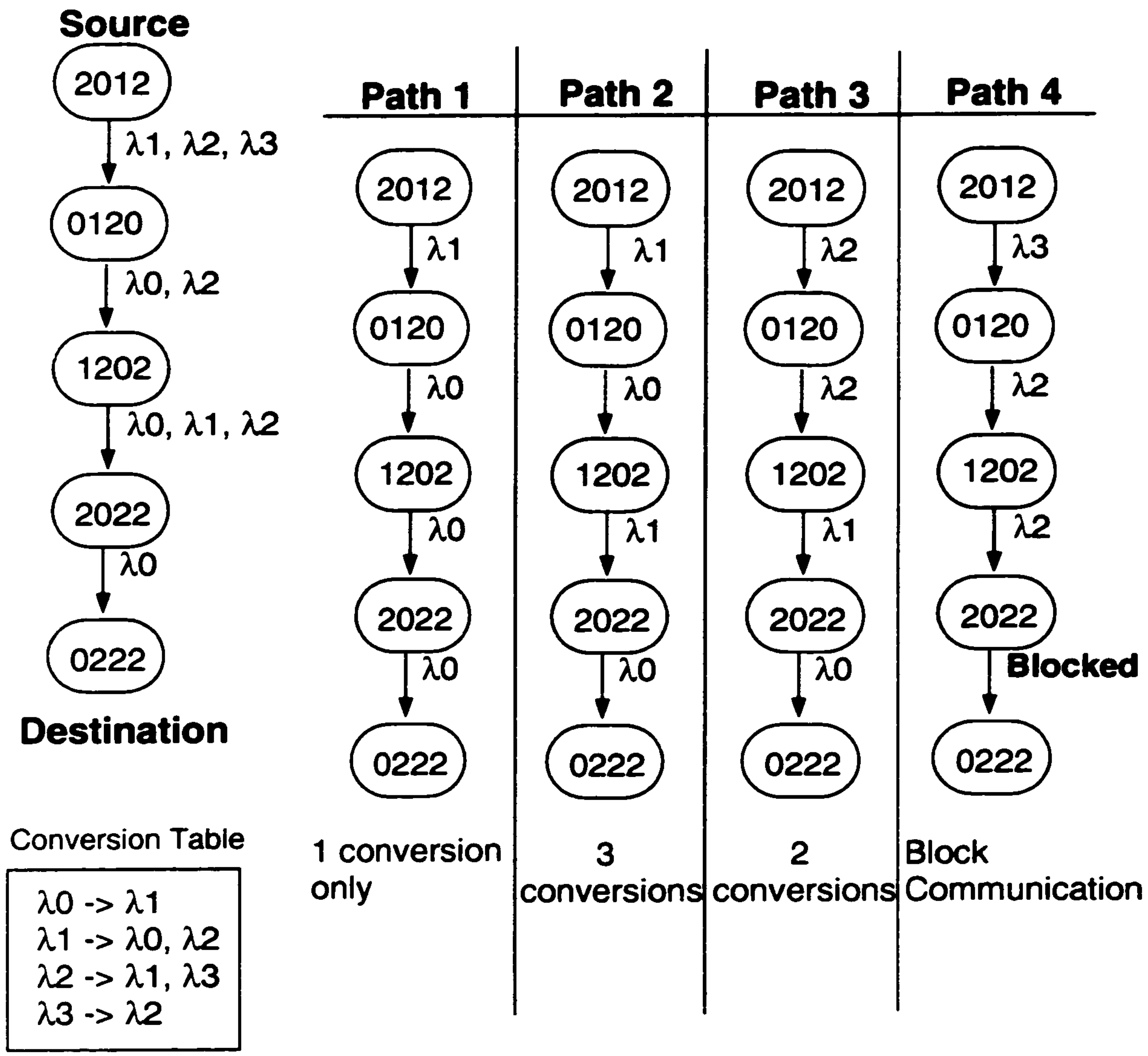


Figure 16 Example 2

Here we examine every path shown above in detail.

For path 1,

1. Node 2012 can use wavelength λ_1 to start sending information to node 0120 at the edge 2012 → 0120.
2. We need to convert from λ_1 to λ_0 because λ_1 is used at the edge 0120 →1202.
3. λ_0 can be used at both edges 1202 → 2022 and 2022 → 0222 because λ_0 is free for both link.
4. This successful communication involves one conversion only.

For paths 2 and 3, they both can obtain successful communications using the same procedures. However, they both need more than one conversion in order to achieve successful communications.

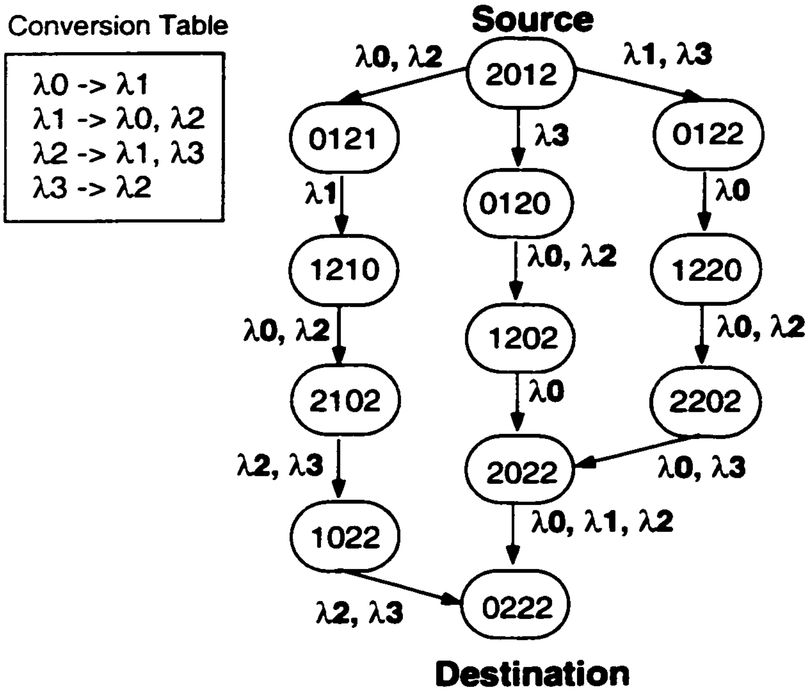
For path 4,

1. We choose wavelength λ_3 to start sending message to node 0120 at the edge 2012 → 0120.
2. We need to convert from λ_3 to λ_2 because λ_3 is used at the edge 0120 →1202.
3. Again, we can remain using λ_2 because λ_2 is free at the edge 1202 → 2022.
4. Since λ_2 is not feasible on the next link 2022 → 0222, we need to convert from λ_2 to λ_1 or λ_3 at the edge. However, none of them is available in this edge. Inevitably, the attempted communication is blocked.

Although path 2 and 3 can be chosen to be the path for transmission, path 1 is favorable because it only involves one wavelength conversion from the source node to the destination node.

Example 3:

This example reveals the benefit of alternative routes in term of the establishment of successful communications. Since the testbed allows users to specify their routing schemes in different topologies, there may be different routes to connect from the source node to the destination node. If the first path is blocked, we can choose another path to try the same communication. Theoretically, in a (Δ, D) de Bruijn graph, it has Δ or $\Delta - 1$ alternative paths of lengths up to $D + 1$ from any source node to any destination node. We repeat the last example and three alternative paths from 2012 to 0222 are shown in Fig. 17.



Path 1		Path 2		Path 3	
Edge	Wavelength	Edge	Wavelength	Edge	Wavelength
2012		2012		2012	
	λ3		λ0		λ1
0120		0121		0122	
	λ2		λ1		λ0
1202		1210		1220	
	Blocked		λ2		λ0
2022		2102		2202	
			λ2		λ0
0222		1022		2022	
			λ2		λ0
		0222		0222	
		2 wavelength conversions		1 wavelength conversion only	

Figure 17 Example 3

When node 2012 starts to establish communication with the destination node 0222, there are three paths we can consider. We look at each of them individually.

1. The first path we consider is the shortest route:

   2012 → 0120 → 1202 → 2022 → 0222

   Unfortunately, this path is blocked because there is no available wavelength at edge 1202 → 2022.

2. The second path is the one where the last digit of the first intermediate node starts with 1:

   2012 → 0121 → 1210 → 2102 → 1022 → 0222

   It is a successful attempt because it can reach from the source node 2012 to the destination node 0222 in two wavelength conversions.

3. The last path is the one where the last digit of the first intermediate node starts with 2:

   2012 → 0122 → 1220 → 2202 → 2022 → 0222

   This is also a successful connection. However, it involves one conversion only, which is one less conversion than that of path 2. Therefore, we choose this path to be our routing path in this example.

As a result, we realize that alternative paths really contribute to the ratio of successful communications if we include alternative paths in the routing schemes we develop in the testbed.

Now we look at our approaches towards different wavelength conversion schemes.

3.4.1 No Wavelength Conversion

Before we discuss our approaches towards different wavelength conversion schemes, we introduce some variables and a function to help us to achieve our goal.

Firstly, we define an array **wavelength_on_edge** to notify the availability of wavelength in a particular edge for the current topology. In other words, it keeps track of whether a wavelength is used up or not. It is originally initialized to be zero. If a wavelength is used in an edge, it is updated to one and cannot be used by another link.

Secondly, a function called **extract_wavelength** is used to check if a wavelength is used up on a particular edge from **wavelength_on_edge**. If so, it returns one to signify a used wavelength. Or the wavelength is available.

Thirdly, for each wavelength λ_i on a particular a → b, we have an index called **Wavelength Conversion Index (WCI)** to represent the numbers of conversions needed to use λ_i at the edge a → b from the source S. Therefore, if WCI (a → b, λ_i) is equal to two, then two conversions are required in order to use λ_i at the edge a → b from the source S. All indices are initialized to be *infinity* ∞.[5] For

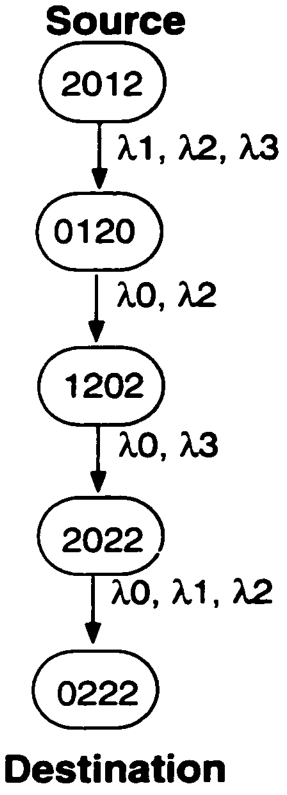
[5] Infinity here stands for a very large number that is greater any number we will use in our simulation since we can not assign infinity in our simulation.

the whole path P, we thus use an array, named **Wavelength Conversion Array**, to keep track of all Wavelength Conversion Indices.

After the testbed generates a source node and a destination node and chooses a path P according to the routing schemes we use, there are as many wavelengths as specified by users in every edge along P. For every wavelength not being used up on each edge belonging to P, we set the wavelength conversion indices to be zero. This can be determined by calling the function extract_wavelength and retrieving the return value from the array wavelength_on_edge. This applies to all edges along P for each intermediate node and the destination node. As a result, we can build up the wavelength conversion array. Next we choose the path that involves a straight set of wavelength from S to D without any conversion from the array. If there is a path found, we record the edge with respective wavelengths used along P. If there is no such path existing in the network, we consider this attempted communication to be blocked.

<u>Example:</u>

We follow the same example all through this thesis. A (3, 4) de Bruijn graph with node 2012 as the source node and node 0222 as the destination node is shown below. The path is also given.



Wavelength Conversion Array

Wavelength Path Edges	λ0	λ1	λ2	λ3
2012 -> 0120	∞	0	0	0
0120 -> 1202	0	∞	0	∞
1202 -> 2022	0	∞	∞	0
2022 -> 0222	0	0	0	∞

Chosen Lighpath

Path Edges	Wavelength
2012 -> 0120	-1
0120 -> 1202	-1
1202 -> 2022	-1
2022 -> 0222	-1

No wavelength conversion

Figure 18 No wavelength conversion

As shown in the diagram, node 2012 is the source node and node 0222 is the destination node with four existing wavelengths. Meanwhile, node 0120, 1202 and 2022 are intermediate nodes. On each edge, there are wavelengths that are available to be used labelled next to the edge. For example, wavelengths λ_1, λ_2

and λ_3 are available for the edge connected to node 2012 and node 0120, whereas there are wavelengths λ_0 and λ_2 available for the edge connected to node 0120 and node 1202, and so on. The wavelength conversion array reveals that the availability of wavelengths along the path P. There are no available wavelengths for four existing wavelengths shown in this example according to the scheme. Therefore this attempt is a blocked communication.

In summary, we specify our approach for no wavelength conversion as follows:

No Wavelength Conversion:

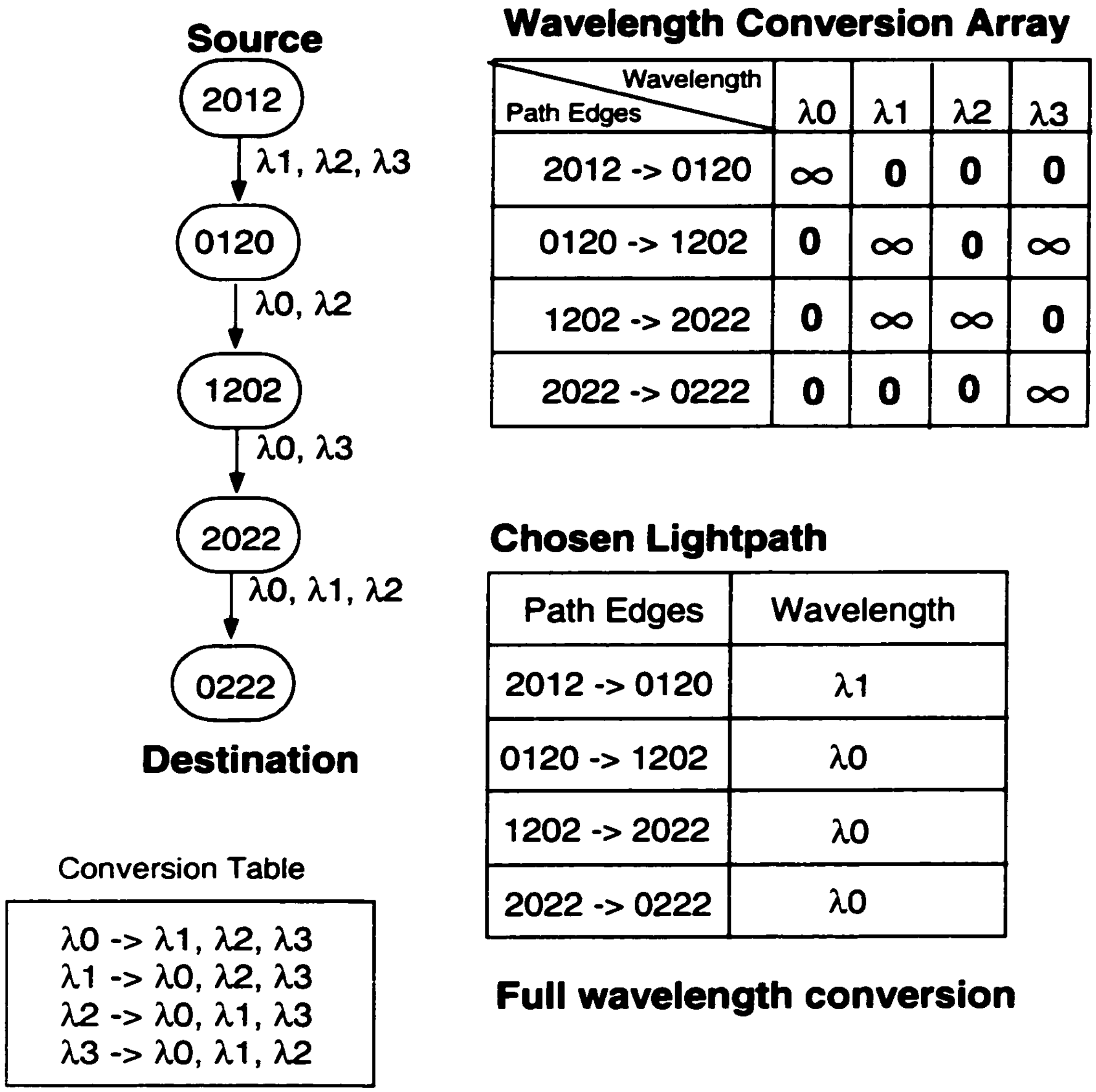
- Update all the indices of all wavelengths that have not been used in the wavelength conversion array used on the path from S to D.
- Choose the lightpath such that all edges in the physical route from S to D have an available channel at the same wavelength.

3.4.2 Full Wavelength Conversion

The technique used for full wavelength conversion is similar to that for no wavelength conversion. After the testbed randomly picks up a source node and a destination node, there are as many wavelengths as specified by users in every edge along path P. For every wavelength not being used up on each edge belonging to P, we set the wavelength conversion indices to be zero. This applies to all edges along P. As a result, we can build up the wavelength conversion array. Next we choose the path that can reach from S to D for whatever wavelength is available from the array. If there is a path found, we record the edge with respective wavelengths used along P. If there is no such path existing in the network, we consider this attempted communication to be blocked.

Example:

A (3, 4) de Bruijn graph with node 2012 as the source node and node 0222 as the destination node is shown below. A conversion table is also given such that we know which wavelength is possible to be converted in the example.



Conversion Table

λ0 -> λ1, λ2, λ3
λ1 -> λ0, λ2, λ3
λ2 -> λ0, λ1, λ3
λ3 -> λ0, λ1, λ2

Wavelength Conversion Array

Wavelength Path Edges	λ0	λ1	λ2	λ3
2012 -> 0120	∞	0	0	0
0120 -> 1202	0	∞	0	∞
1202 -> 2022	0	∞	∞	0
2022 -> 0222	0	0	0	∞

Chosen Lightpath

Path Edges	Wavelength
2012 -> 0120	λ1
0120 -> 1202	λ0
1202 -> 2022	λ0
2022 -> 0222	λ0

Full wavelength conversion

Figure 19 Full wavelength conversion

The diagram reveals that node 2012 is the source node, node 0222 is the destination node, and there are four existing wavelengths in the network. Meanwhile, node 0120, 1202 and 2022 are intermediate nodes. The wavelength conversion array reveals that the availability of wavelengths along the path P. Since there is no straight set of wavelength that can be used in this example, we consider wavelength conversion according to the conversion table shown in the diagram to establish the path. There are a number of paths available for a successful communication. Here we choose λ_1 for the edge 2012 → 0120 and λ_0 for the remaining edges 0120 → 1202, 1202 → 2022 and 2022 → 0222. Since this path is the first successful attempt, we choose this path as our routing path.

Now we conclude our approach for full wavelength conversion as follows:

Approach for Full Wavelength Conversion:

- Update all the indices of all wavelengths that have not been used in the wavelength conversion array used on the path from S to D.
- Choose the lightpath that is able to convert from one wavelength to another from S to D if necessary.

3.4.3 Limited Wavelength Conversion

How we derive a path in limited wavelength conversion scheme can be split into two parts. The first part is the same as the other two conversion schemes. In other words, for every wavelength not being used up on each edge belonging to the path P, we set the wavelength conversion indices to zero. We can thus build up the wavelength conversion array. The second part is divided into three processes. They are evaluated at the source node, any intermediate node and the destination node, respectively [25].

Before we discuss their algorithms, we assume that the path P consists of k edges and can be represented in the form of $x_0 = S \rightarrow x_1 \rightarrow \ldots \rightarrow x_k = D$ where S is the source node, D is the destination node and $x_1 \rightarrow \ldots \rightarrow x_{k-1}$ are the intermediate nodes.

Here the algorithm describes the process evaluated at the source node S:

Processing at the source node x_0 = S:

1. Select a particular path P from S to D depending on the routing schemes we use.
2. If every Wavelength Conversion Index WCI ($x_0 \rightarrow x_1$, λ_i) = ∞, report that this connection is blocked using the path.
3. If not, continue onto the next process.

Now we evaluate the process at the intermediate nodes. At each intermediate node, we update the wavelength conversion index of each wavelength λ_i on the outgoing edge. The value can be either the index of the same wavelength λ_i on its incoming edge, or the index of another wavelength λ_j, convertible to λ_i, incremented by 1. The value of the index should be the lowest possible value among those indices.

The following algorithm explains the process at the intermediate nodes:

```
Processing at the intermediate nodes $x_t$ where $0 \leq t < k - 1$:

1. For every $\lambda_i$ not used on the outgoing edge $x_t \rightarrow x_{t+1}$,
    a. If WCI $(x_{t-1} \rightarrow x_t, \lambda_i) < \infty$,
       Assign WCI $(x_{t-1} \rightarrow x_t, \lambda_i)$ to WCI $(x_t \rightarrow x_{t+1}, \lambda_i)$.
    b. For every $\lambda_j$ not used on the incoming edge $x_{t-1} \rightarrow x_t$,
        • If $\lambda_j$ is convertible to $\lambda_i$, and
        • If WCI $(x_{t-1} \rightarrow x_t, \lambda_j) + 1 < Min (\lambda_i)$,
          where Min stands for minimum transition value from $\lambda_j$ to $\lambda_i$ and
          it is originally initialized to be $\infty$.

          Assign WCI $(x_{t-1} \rightarrow x_t, \lambda_j) + 1$ to Min $(\lambda_i)$
        • Finally assign Min $(\lambda_i)$ to WCI $(x_t \rightarrow x_{t+1}, \lambda_i)$ after every $\lambda_j$ is
          considered.
2. If every WCI $(x_j \rightarrow x_{j+1}, \lambda_i) = \infty$,
   this attempt is blocked using the path.
```

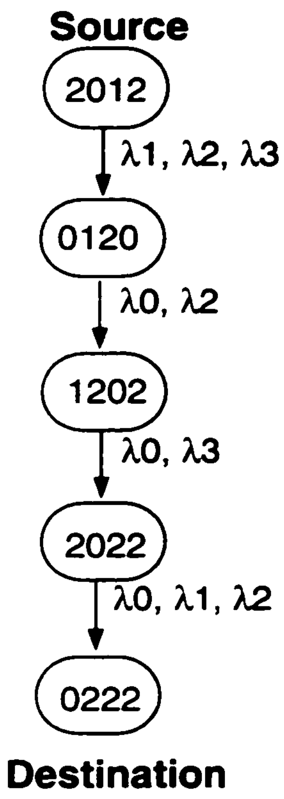
At destination node D, we look for path P and the vector of wavelengths W, which consists of the lowest number of wavelength conversions possible, according to the strategy we adopt as follows:

```
    Processing at the destination node $x_k = D$:
1.  At the edge $x_{k-1} \rightarrow x_k$,
    a.  Select the minimum WCI $(x_{k-1} \rightarrow x_k, \lambda_i)$.
    b.  Assign $\lambda_i$ to W [k].
2.  For every other edge $x_{t-1} \rightarrow x_t$ on P,
    a.  If WCI $(x_{t-1} \rightarrow x_t, W[t])$ = WCI $(x_t \rightarrow x_{t+1}, W[t+1])$,
        Assign W [t + 1] to W [t].
    b.  Or choose the minimum WCI $(x_{t-1} \rightarrow x_t, \lambda_j)$ where $\lambda_j$ is convertible
        to $\lambda_i$,
        Assign $\lambda_j$ to W [t].
```

Finally we choose the path that has the lowest number of conversions along the path from the array. If a path is found, we record the edge with the respective wavelengths used along P. If there is no such path existing in the network, we consider this attempted communication to be blocked.

Example:

A (3, 4) de Bruijn graph with node 2012 as the source node and node 0222 as the destination node is shown below. Another conversion table is also accompanied to express the possibility of wavelength conversion in the example.



Wavelength Conversion Array

Path Edges \ Wavelength	λ0	λ1	λ2	λ3
2012 -> 0120	∞	0	0	0
0120 -> 1202	1	∞	0	∞
1202 -> 2022	1	∞	∞	1
2022 -> 0222	1	2	2	∞

Chosen Lightpath

Path Edges	Wavelength
2012 -> 0120	λ1
0120 -> 1202	λ0
1202 -> 2022	λ0
2022 -> 0222	λ0

Conversion Table

λ0 -> λ1
λ1 -> λ0, λ2
λ2 -> λ1, λ3
λ3 -> λ2

Limited wavelength conversion

Figure 20 Limited wavelength conversion

First, we update those available wavelengths in the wavelength conversion array. Next, as we go to the second part, we choose path P according to our algorithms as follows:

1. Since wavelengths λ_1, λ_2 and λ_3 are not in use at the source node 2012 for the edge 2012 → 0120, we continue the next process.

2. For the intermediate nodes, we consider them sequentially:

   a. For the edge 0120 → 1202,

      we assign 1 to WCI (λ_0) because

      - λ_0 is not available on the incoming edge 2012 → 0120, and
      - the index of λ_1 on 2012 → 0120, after being incremented by 1, is less than the *Min* (λ_0), originally set to be 0, on 0120 →1202. Therefore we consider wavelength conversion from λ_1 to λ_0.

      WCI (λ_2) = 0 because

      - λ_2 is not used on the incoming edge 2012 → 0120.
      - λ_2 is available on the outgoing edge 0120 → 1202.

   b. For the edge 1202 → 2022,

      WCI (λ_0) remains to be 1 because

      - λ_0 is available on the incoming edge 0120 → 1202.
      - λ_0 is also available on the outgoing edge 1202 → 2022.

WCI (λ_3) is 1 because

- λ_3 is not available on the incoming edge 0120 → 1202, and
- the index of λ_2 on 1202 → 2022, after being incremented by 1, is less than the *Min* (λ_3) on 1202 → 2022. Therefore we consider wavelength conversion from λ_2 to λ_3.

3. At the destination node 0222 for the edge 2022 → 0222,

WCI (λ_0) remains to be 1 because

λ_0 is available on the incoming edge 1202 → 2022.

WCI (λ_2) is 2 because

a. λ_2 is not available on the incoming edge 1202 → 2022, and

b. the index of λ_0 on 1202 → 2022 after being incremented by 1 is less than the *Min* (λ_1) on 2022 → 0220. Therefore we consider wavelength conversion from λ_0 to λ_1.

WCI (λ_3) is 2 because

a. λ_2 is not available on the incoming edge 1202 → 2022, and

b. the index of λ_3 on 1202 → 2022, after being incremented by 1, is less than the *Min* (λ_2) on 2022 → 0220. Therefore, we consider wavelength conversion from λ_3 to λ_2.

Going backward through the edges of path P, we look for the wavelengths that will form a lightpath with the lowest number of wavelength conversions as possible. First, we select λ_0 on the edge 2022 → 0220 because WCI (λ_0) involves the lowest number of conversions among all the indices on that edge. Next the WCI (λ_0) on 1202 → 2022 is equal to the WCI (λ_0) on the edge 2022 → 0222; thus we choose λ_0. This is the same case for λ_0 on the edge 0120 → 1202. At last, we evaluate the edge 2012 → 0120, but the WCI (λ_0) on 1202 → 2022 is not equal to the WCI (λ_0). We search for a wavelength that is convertible to λ_0. From the wavelength conversion array and the conversion table, we know λ_1 has the minimum number of wavelength conversion and is the only wavelength capable of conversion. As a result, we choose λ_1 on the edge 2012 → 0120 and complete the routing path with one wavelength conversion.

Here we summarize our approach for limited wavelength conversion as follows:

Approach for Limited Wavelength Conversion:

- Update all the indices of all wavelengths that have not been used in the wavelength conversion array used on the path from S to D.
- Choose the lightpath that has the lowest number of conversions along the path.

Chapter 4 NETWORK SIMULATION

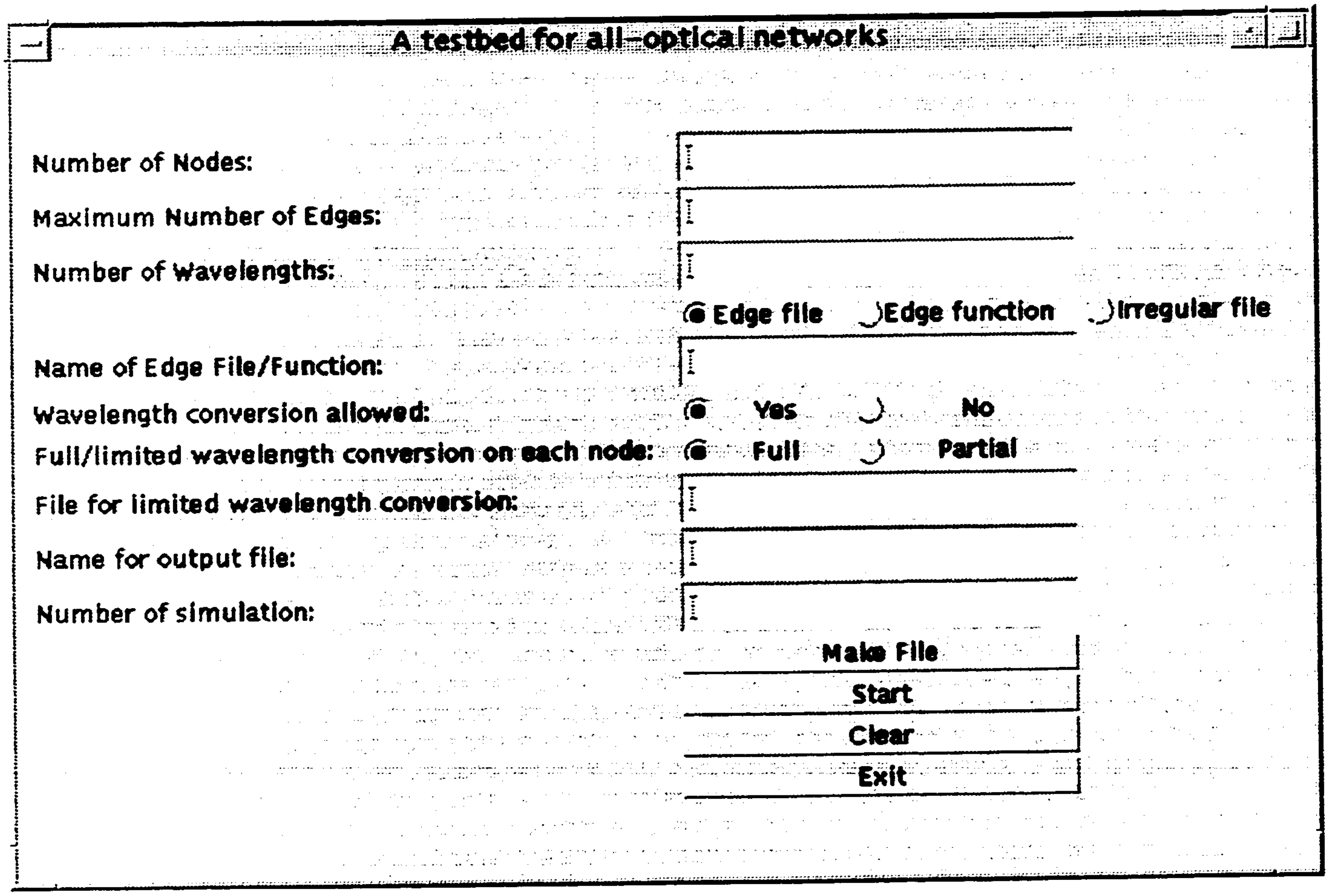
In this thesis, we will investigate our problem using simulation generated by the testbed. Before we go further, we first give an introduction to our testbed simulation and describe how we have implemented the simulation processes. In addition, in appendix A, we will give an introductory instruction of running the testbed.

4.1 Instruction for Manipulation of the Testbed Simulation

This testbed consists of two parts: a user interface and a simulation program. The user interface is created in Java whereas the simulation program is built in C. When a user finishes entering all textfields and choosing all option buttons in the user interface, the user should click the buttons to start the simulation. Here we use Java's native method to invoke a C program to run the simulation program. Afterwards, results are written back into a file specified by the user.

Here we will take a look at the testbed created for this thesis.

Figure 21 An interface for testbed



A testbed for all-optical networks

Number of Nodes:

Maximum Number of Edges:

Number of Wavelengths:

● Edge file ○ Edge function ○ Irregular file

Name of Edge File/Function:

Wavelength conversion allowed: ● Yes ○ No

Full/limited wavelength conversion on each node: ● Full ○ Partial

File for limited wavelength conversion:

Name for output file:

Number of simulation:

Here we explain different fields in the interface as follows:

Number of Nodes	Number of nodes in the topology.
Maximum Number of Edges	Maximum number of edges this topology has
Number of Wavelengths	Number of wavelengths considered
Number of transceivers	How many transceivers in each node
Edge File/Edge Function	Are we given the edge file or edge function in the topology?
Name of Edge File/Edge Function	Filename for edge file or edge function
Wavelength conversion allowed	Do we allow wavelength conversion?
Full/limited wavelength conversion	Are we using full wavelength conversion or limited wavelength conversion?
File contains information about limited wavelength conversion	The file to store all information about wavelength conversion in every node

Table 2 Different fields in the interface (Continued) . . .

Name for output file	The file to store all hits and misses from the simulation
Number of simulation	Number of time the simulation runs
"Make File"	To create the "makefile" to link all programs together
"Start"	To start the simulation
"Clear"	To clear all values
"Exit"	To exit the testbed

Table 2 Different fields in the interface

4.2 Simulation processes

In this section, we are going to explain how the simulation works in detail.

4.2.1 Introduction

Initially, the simulation begins with the network having no communications at all. A random generation of connections between source and destination begins and establishes a lightpath if there is a successful connection. All attempts to establish connections at this early stage are undoubtedly successful. A wavelength is thus used up after a successful communication. However, as simulation goes on, some of these attempts succeed and some fail. There are various reasons for failure such as no available wavelength and lack of transmitters and receivers.

Finally, the network reaches the "saturated" stage where no more connection is possible. We stop our simulation program and collect the output information, including the number of successful connections (hits) and the number of failed ones (misses).

Since we want to know the importance of wavelengths used in the network, we vary the total number of wavelengths in the network in different experiments while we keep other variables constant throughout each experiment.

4.2.2 Initialization of Parameters

There are a list of parameters we need to initialize before the simulation takes place:

1. Number of simulation.
2. Number of nodes.
3. Maximum number of edges.
4. Number of wavelengths.
5. Number of transceivers per node.
6. Edge file or edge function.
7. Wavelength conversion array.

4.2.3 Random Generation of a Valid Communication Between Source and Destination

As the simulation begins, it will randomly generate a source-destination pair using the following algorithm:

```
Random Generation of a Source and a Destination
Repeat
    Pick up a random source node named S
Until S has an available transmitter
Repeat
    Pick up a random destination node named D
    If D has no available receiver, discard D.
    If D and S are the same node, discard D.
    If a previous connection between S and D failed, discard D.
Until D satisfies all conditions.
```

4.2.4 The Attempt to Create Lightpath Using Available Wavelength

After the simulation generates a source and a destination, it tries to establish a connection between them using the methodology given below.

When there is a path P from the source node S to the destination node D, this path usually means that it has k edges and has the pattern $x_0 = S \rightarrow x_1 \rightarrow$

$x_2 \rightarrow \ldots \rightarrow x_k = D$, where the first node is the source node S, the last node is the destination node D and $x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_{k-1}$ are the intermediate nodes.

In our algorithm, for each wavelength λ_i on a particular edge a → b, we keep an index called Wavelength Conversion Index (WCI) to represent the number of conversions needed to use λ_i at the edge a → b from the source S. In addition, we let N be the total number of wavelengths available in the network and a wavelength λ_i be $0 \leq i < N$.

We recall the different approaches to establish a lightpath in three wavelength conversion schemes in section 3.4:

No Wavelength Conversion:

- Update all the indices of all wavelengths that have not been used in the wavelength conversion array used on the path from S to D.
- Choose the lightpath such that all edges in the physical route from S to D have an available channel at the same wavelength.

Full Wavelength Conversion:

- Update all the indices of all wavelengths that have not been used in the wavelength conversion array used on the path from S to D.
- Choose the lightpath that is able to convert from one wavelength to another from S to D if necessary.

Limited Wavelength Conversion:

- Update all the indices of all wavelengths that have not been used in the wavelength conversion array used on the path from S to D.
- Choose the lightpath that has the lowest number of conversions along the path.

Finally, we record the path according to different wavelength conversion schemes. The vector of wavelength W specifies the particular wavelength the path uses in each edge.

Establishing the Lightpath:

1. Starting from source node S, for every node $x_t \neq$ D on the path P, we record W [t].
2. Decrement the number of transceivers for every node used in P.

Chapter 5 RESULTS OF SIMULATION EXPERIMENTS

In order to study the effect of wavelength conversion in all-optical networks based on different topologies mentioned in section 2.7, we vary relevant parameters and determine how they affect the network performance. We set up the parameters according to the input specified by the users in the Java interface mentioned in section 4.1. The parameters include the following:

1. Number of nodes.
2. Maximum number of edges.
3. Number of wavelengths.
4. Number of transceivers per node.
5. Wavelength conversion schemes.
6. Edge file or edge function.
7. Number of simulation.
8. Name for output file.

We have explained the parameters in Chapter 4. However, there are significant factors for the parameters to affect the network performance if we alter the parameters.

1. If the number of wavelengths available increases, we will see more successful connections in our simulation. However, due to cost considerations, we look for the minimum number of wavelengths.
2. If the number of transceivers per node increases, more successful connections will be possible because we allow more attempted communication in each node.
3. Usually, full wavelength conversion generates the lowest blocking probability, limited wavelength conversion has lower blocking probability, and no wavelength conversion comes last.
4. The simulation will be more accurate if the number of simulation increase.

After all combinations of those parameters, we want to determine how we may search for the best performance with the least cost.

As a result, we collect the total number of hits and misses from the simulation, and we thus calculate the blocking probability of the network by the ratio of the number of blocked communications to the total number of attempted communication.

In our experiments, we will run the simulation ten times for each network even though we can change the number of simulations. Originally, we intended to run the simulation 100 times in each topology. However, we find out that having 100 trials takes so much time and uses up a lot of computer resources compared to having 10 trials. They both come to the same conclusion, although 100 trials definitely produces a more accurate result. We can look at Fig. 22 and see their difference.

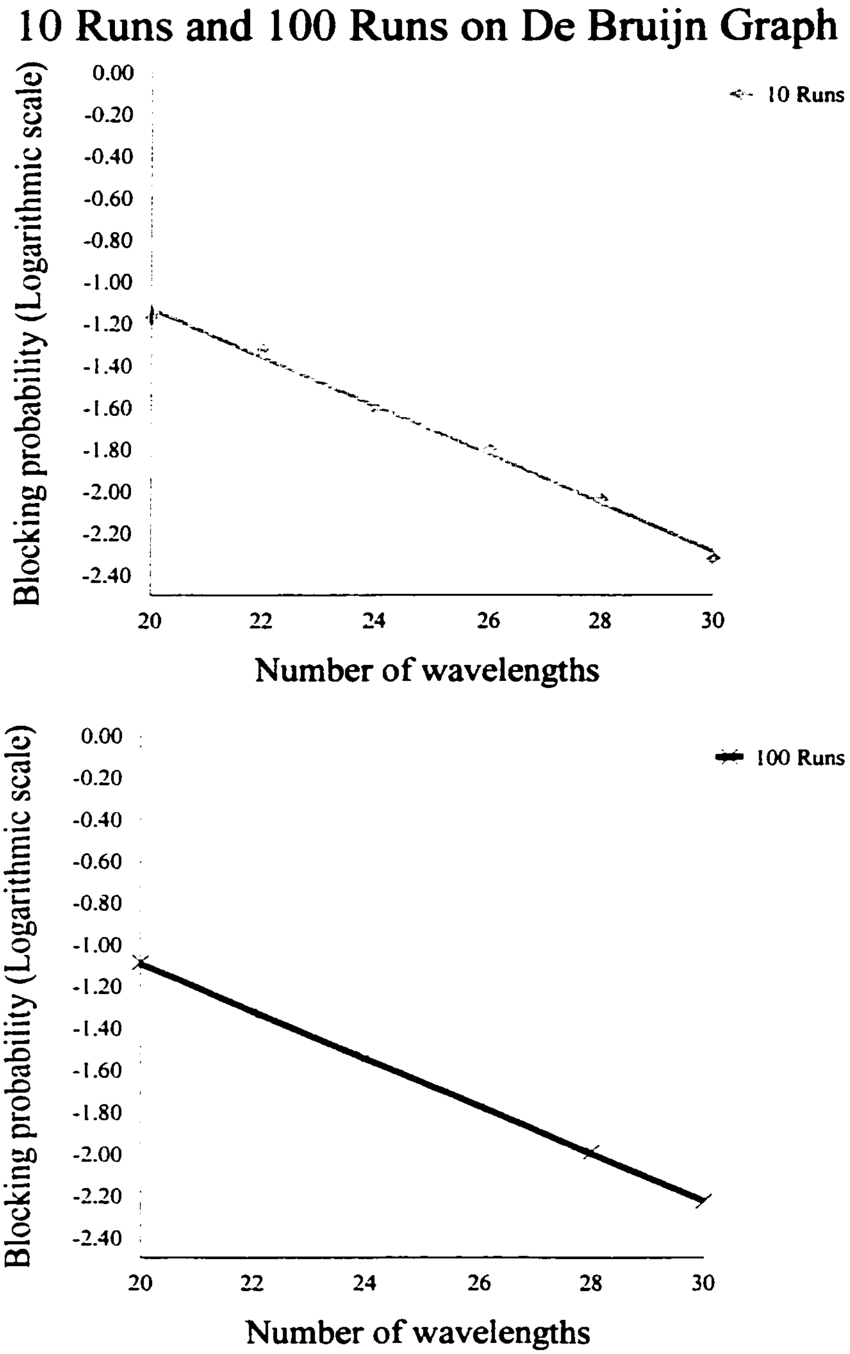


Figure 22 Charts of 10 Runs Vs. 100 Runs

5.1 Experiments

We have used different networks described in section 2.7 as our topologies and implemented a comprehensive series of experiments using different set of data. The objectives of our experiments are listed below:

1. In experiment 1:

   to study the effect of wavelength conversion on the Ring Network.

2. In experiment 2:

   to study the effect of wavelength conversion on the De Bruijn Graph.

3. In experiment 3:

   to study the effect of wavelength conversion on the Manhattan Street Network (MSN).

4. In experiment 4:

   to study the effect of wavelength conversion on an arbitrary irregular topology.

All our experiments involve three sets of data, namely full, limited and no wavelength conversion. In each experiment, results on no wavelength conversion serves as a main case for our comparison. Here we illustrate our experimental results in detail.

5.1.1 Experiment 1

Set of parameters:

1. It has 300 nodes.
2. Range of number of wavelength to be used is from 300 to 650 with intervals of 50.
3. Number of connection is 5.
4. Maximum edge length is 300.
5. We have created an edge file for the ring network to specify the structure of the network.
6. There is a routing function for the ring network we have adopted as mentioned before.

The chart of this experiment is given in Fig. 23.

Observation: As we increase the number of wavelengths in the network, we can see a dramatic drop of blocking probability at the range of 500 — 600 wavelengths. It can create more successful communication when the number of wavelengths reach that point. One interesting discovery in this experiment is that limited wavelength conversion almost obtains the same result as full wavelength conversion, even though limited wavelength conversion can only provide a conversion range of one wavelength.

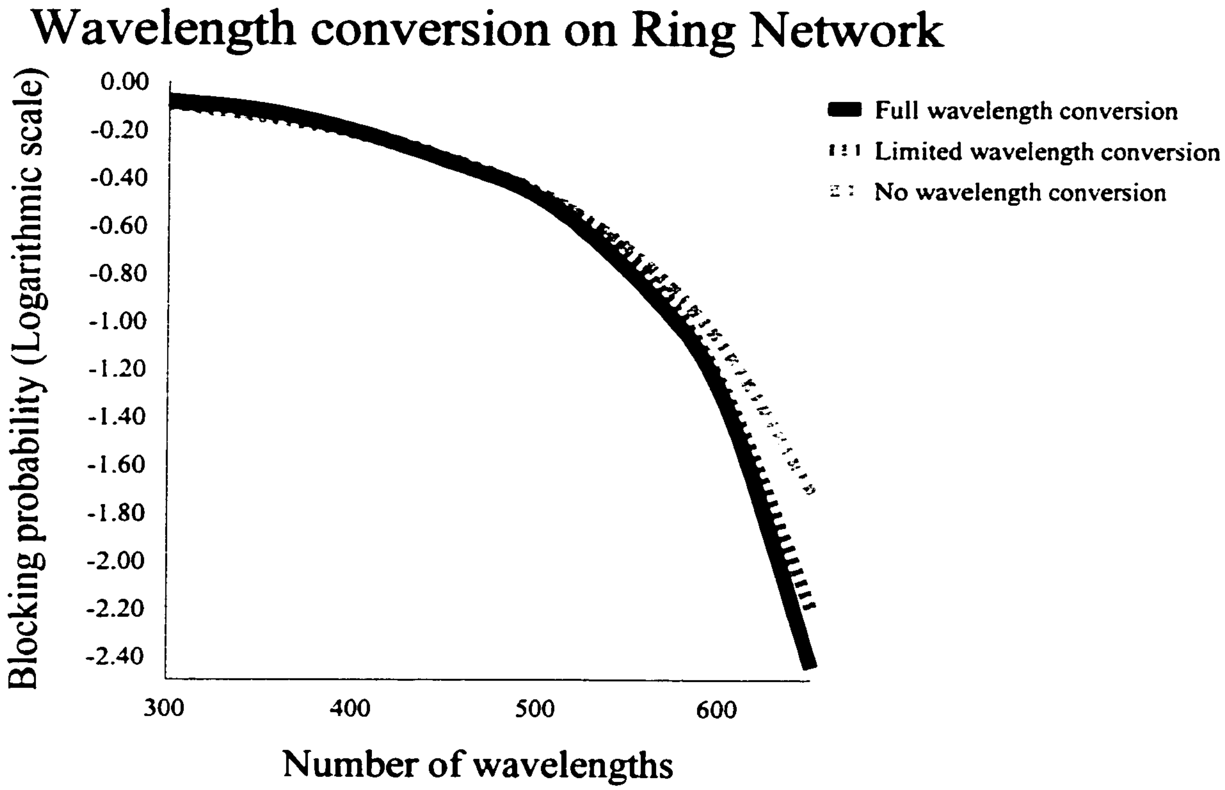


Figure 23 Effect of wavelength conversion on the Ring Network

5.1.2 Experiment 2

Set of parameters:

1. It has 1024 nodes, i.e. a (4,5) de Bruijn graph.
2. Range of number of wavelengths to be used is from 20 to 30 with 1 as the interval.
3. Number of connections is 10.
4. Maximum edge length is 30.
5. We have established an edge function for the de Bruijn graph. The function is able to find the edge connecting any two nodes if there is a communication between the nodes.
6. A routing function for the de Bruijn graph is also created.

The chart of this experiment is given in Fig. 24.

Observation: As we increase the number of wavelengths in the network, we can see a linear improvement over the network in terms of blocking probability. Limited wavelength conversion is also as good as full wavelength conversion in this network. Another interesting note we can find in this experiment is that wavelength conversion can boost up the performance compared to a network with no wavelength conversion. In other words, wavelength conversion provides a huge difference with respect to the performance of no wavelength conversion for the same network.

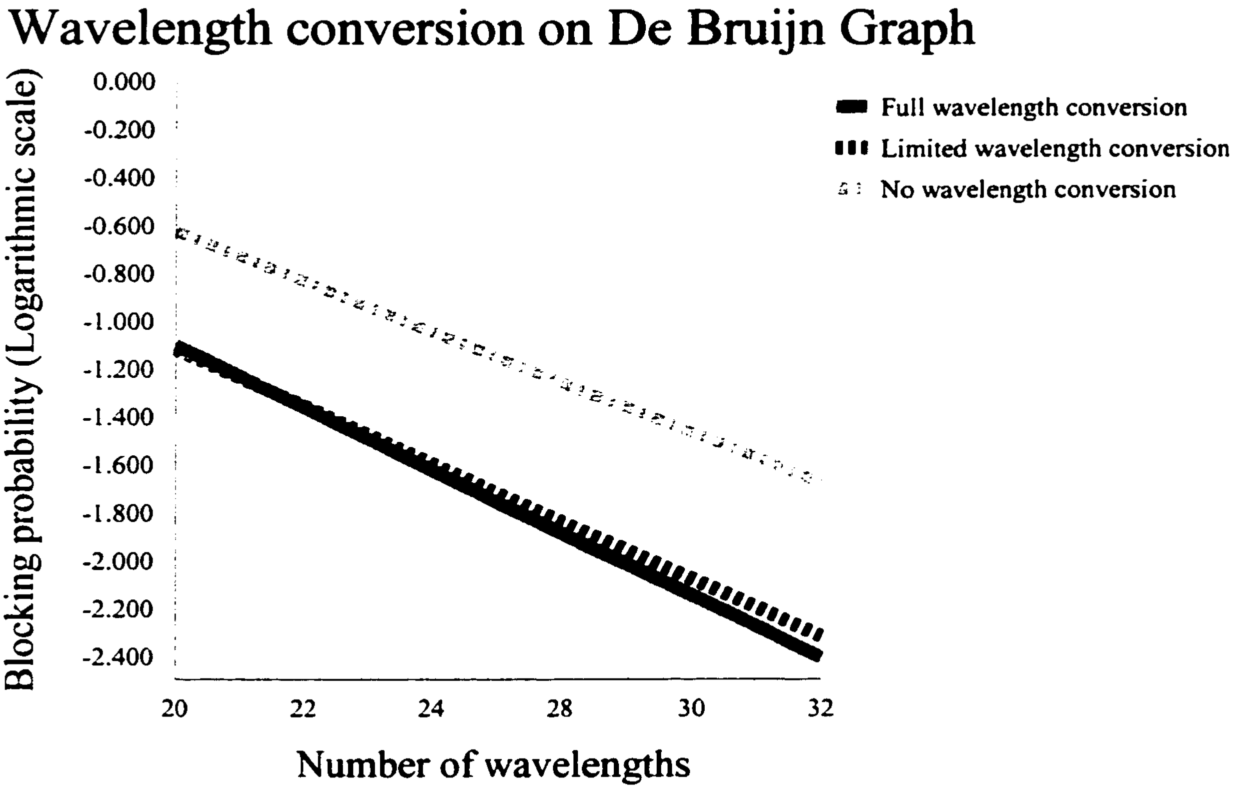


Figure 24 Effect of wavelength conversion on the De Bruijn Graph

5.1.3 Experiment 3

Set of parameters:

1. It has 900 nodes.
2. Range of number of wavelengths to be used is from 100 to 240 with 20 as the intervals.
3. Number of connection is 20.
4. Maximum edge length is 80.
5. An edge function similar to the one generated for the de Bruijn graph has been generated.
6. A routing function especially made for the Manhattan Street network is created.

The chart of this experiment is given in Fig. 25.

Observation: This is the only experiment where full wavelength conversion can excel limited wavelength conversion in a great deal in terms of blocking probability. Limited wavelength conversion can only give a slight improvement in a linear pattern, whereas full wavelength conversion is a quadratic shape with several drop points.

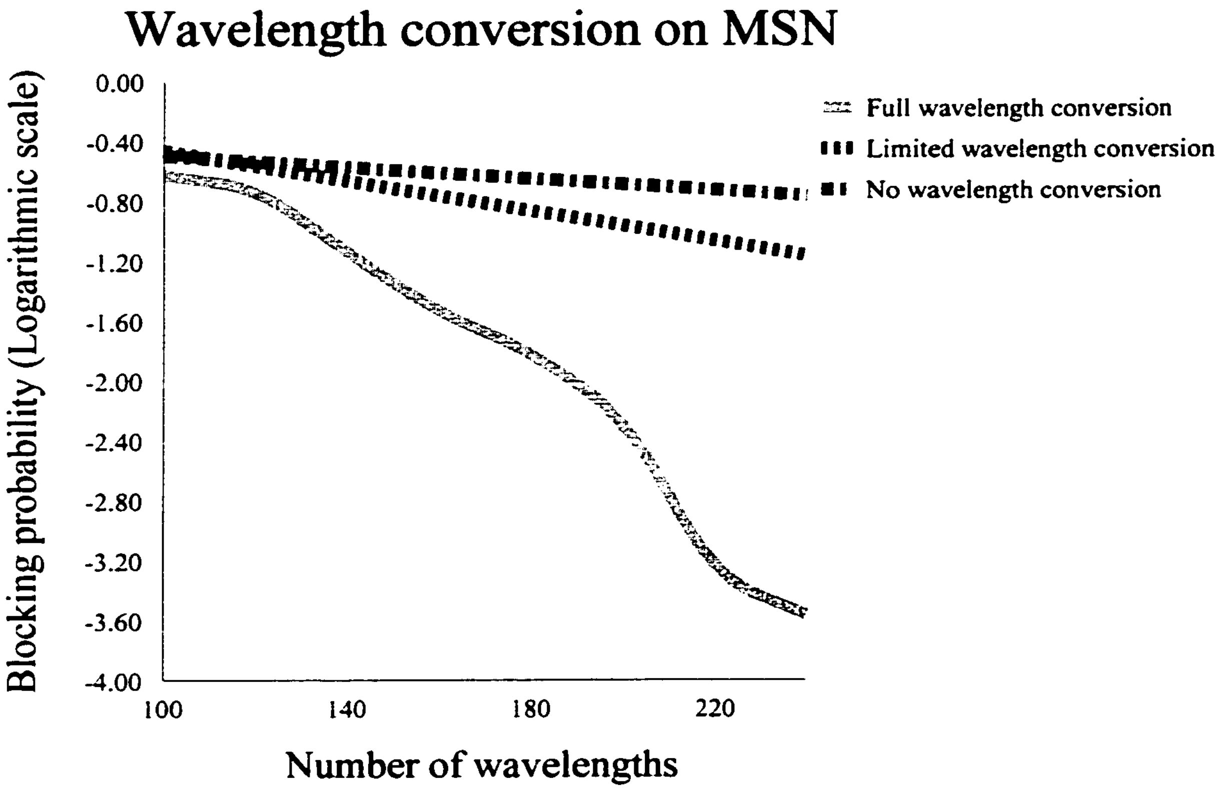


Figure 25 Effect of wavelength conversion on the Manhattan Street Network

5.1.4 Experiment 4

Set of parameters:

1. It has 100 nodes.
2. Range of number of wavelengths to be used is from 100 to 600 with 50 as interval
3. Number of connections is 10
4. Maximum edge length is 100
5. Instead of using an edge function, we adopt an edge file where the file specifies a communication between any two nodes with a weight value.
6. We use Dijkstra's shortest-path algorithm for our routing scheme.

The chart of this experiment is given in Fig. 26.

Observation: Due to the irregular connectivity pattern in the topology, there is no difference when applying wavelength conversion because there may not be a path connecting the source node and the destination node at all. Sometimes, even the blocking probability for full wavelength conversion is higher than that for no wavelength conversion. This is because of the higher probability of failed communication generated from randomization in the testbed for the source-destination pairs used in full wavelength conversion. Therefore, we cannot see much improvement in the topology we use for the irregular topology. However, we will not conclude that wavelength conversion does not provide efficient result for any arbitrary irregular topology because we will not know the connectivity pattern for those topologies until we analyze each of them. We aim to prove that we are able to accept an irregular topology to be in our experiment. We will not predict the effect of wavelength conversion on an arbitrary irregular topology in our judgement.

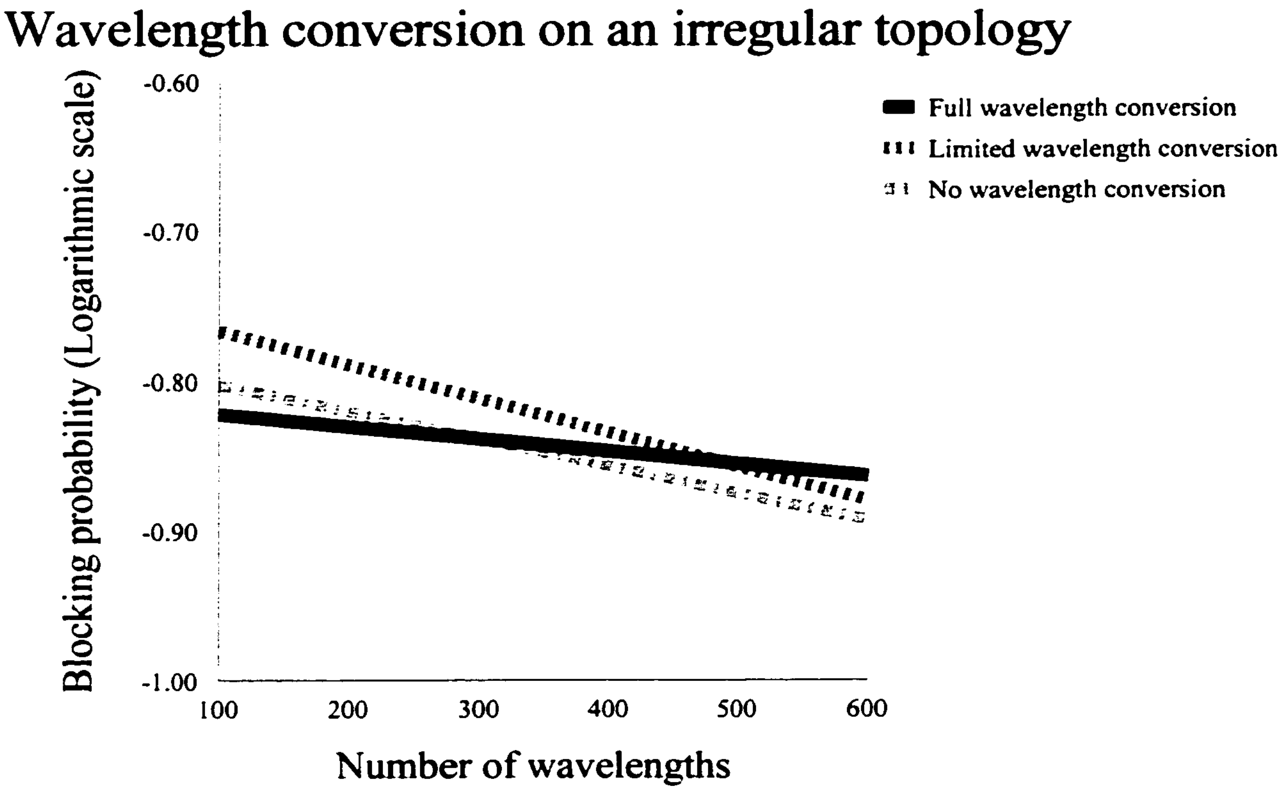


Figure 26 Effect of wavelength conversion on an irregular topology

5.2 Critical Summary

To conclude all the experiments, we have achieved the build up of a testbed for all-optical networks and using the testbed to analyze the blocking probability of each network we mentioned in Section 2.7. One general observation is that employing wavelength conversion in a network does produce a beneficial effect compared to not having wavelength conversion in the same network. In other words, wavelength conversion can produce a lower blocking probability for the same number of wavelengths used in the network. It can minimize the number of wavelengths used in the network in order to obtain an acceptable rate of blocking probability 10^{-3}.

Using wavelength conversion can cause additional network overhead. Since more resources are used for searching a successful lightpath with available wavelengths and keeping track of the wavelength along the lightpath, this limits our simulation in terms of time and computer resources available. Usually our simulation finishes within a day. However, for an irregular topology, the simulation takes longer to complete because we have to establish the shortest path for every node every time a source-destination pair is randomly generated. Therefore, there are additional expenses to be considered if wavelength conversion takes place.

Another important observation in the experiments is that limited wavelength conversion gives the almost-identical performances compared to full wavelength conversion. In other words, increasing the conversion range does not have very much impact with respect to the overall performance of the network. Our observation agrees with [23].

Chapter 6 FUTURE WORK

Even though we implement this testbed generally enough to be used for wavelength conversion in various networks, we find it difficult to allow certain flexibilities. In order to keep our work to be manageable, there are some restrictions that we follow and we look for some additional work to be done in this area in future.

In this thesis, we assume that every node is equipped with same number of wavelength converters. However, as we know that there are some nodes more frequently visited than others, it creates a bottleneck problem. It results in a high blocking probability even though there are enough wavelengths available in the network. Therefore, we suggest that there can be an arbitrary placement of wavelength converter in the network so that the nodes having more traffic can have more wavelength converters in order to avoid blocked communications.

In our simulation, we assume that calls are established and they continue until the next trial. However, we should expect that every connection starts and finishes at certain period of time. Therefore, we will need to investigate this restriction.

A source node and a destination node are usually generated under random generation and based on uniform distribution. However, in reality, we expect a pattern of communication implemented by a set of probability. In other words, there may be certain patterns of calls generated within particular subsets of nodes.

On the other hand, some other nodes are having fewer chances of being either the source node or the destination node. To our extent, it will be beneficial having such a feature in the testbed.

Besides wavelength conversion, another important topic we are interested in providing a lower blocking probability is alternate path routing [19]. As we know, alternate paths can solve bottleneck problems by providing another path starting at the source node and ending at the destination node.

Chapter 7 CONCLUSION

The purpose of this work was to develop a simulator for all-optical networks allowing an user to define his/her own topology, routing scheme and wavelength conversion scheme. We accomplished this using a simulator written in C and having a frontend user interface written in Java. To test whether it is convenient to use the testbed, we considered a hypothesis regarding wavelength conversion and tested it using a variety of network topologies. Our hypothesis was that limited wavelength conversion, in general, is as good as full wavelength conversion. To test this hypothesis, we defined three different topologies — the ring, the torus and the de Bruijn graph topologies. We defined our own routing functions for each of these topologies and tested the blocking probabilities using no wavelength conversion, full wavelength conversion and limited wavelength translation. We found that it was very easy to carry out such a comparative study. The result of the experiment confirmed the hypothesis regarding limited wavelength conversion postulated by Yates [23]. This means that using available technology [23], all-optical networks with limited wavelength conversion may be constructed with all the benefits of full wavelength conversion.

We also repeated the experiment using an irregular topology where it was necessary to define each edge of the network individually. Again, it was simple to use our testbed for irregular topologies. We conclude that it is very convenient to use this testbed to look at new fault avoidance schemes, test new architectures, and new routing schemes.

APPENDIX A

Instruction to run the testbed using De Bruijn Graph as an example:

In order to illustrate the use of the testbed, we give a step-by-step example to explain the procedures involved with diagrams as snapshots. We intend to use the (4, 5) De Bruijn Graph as our example because we want to keep our example consistent with the same approach. In addition, we employ limited wavelength conversion schemes in the testbed so that we can identify the wavelength conversion scheme we use.

Step 1: Initialization

There are a few initialization we have to consider before we run the testbed.

1. Do we have a routing function?
2. Do we have an edge function or an edge file already created?
3. Do we also have header files for those two functions mentioned above so that the testbed can include them when we compile it?
4. Do we have a file with information about limited wavelength conversion? For example, which wavelength can be converted into another wavelength?

In this example, we have a routing function called **route_debruijn.c** and an edge function called **define_debruijn_edge.c**.

If all the answers are satisfied, we can proceed to modify the simulation program **simulation.c** written in C. The reason for modification is that we need to change the functions when we call for a routing scheme and define the edge in the De Bruijn Graph. Furthermore, we need to know the information about wavelength conversion before we actually run the simulation. After all the modifications are done, we can run a user interface built by Java. The commands for popping up the interface are given as follows:

How to run the Java Frontend

- javac Frontend.java

This command compiles the Java Frontend program.

- javah -jni NativeMethod

Here we use the command **javah** to create a JNI-style header file (a .h file) from **NativeMethod** Java class in **Frontend.java**. This header file is machine-generated and contains a function definition for the implementation of the native method **start_simulation()** defined in **NativeMethod** Java class. We invoke the function **start_simulation()** to communicate with **simulation.c** when the simulation starts.

- cc -G -I /usr/java/include -I /usr/java/include/solaris \ NativeMethodImp.c -o libstartsim.so

Now we design what the native method does in a source file **NativeMethodImp.c**. The implementation will be a typical function that is invoked when the Java method **start_simulation()** initiates. After we write the native method implementation, we are ready to create a shared library **libstartsim.so**. Using the command above, compile the **NativeMethod.h** and **NativeMethodImp.c** into the shared library so that **Frontend.java** can be referred to. In addition, we need to set up a library path so that the Java runtime system is able to search for the right directory while loading libraries. The path is where the shared library **libstartsim.so** locates using the following command:

setenv LD_LIBRARY_PATH /home/ucc/lama/60797/experiment1

- java Frontend

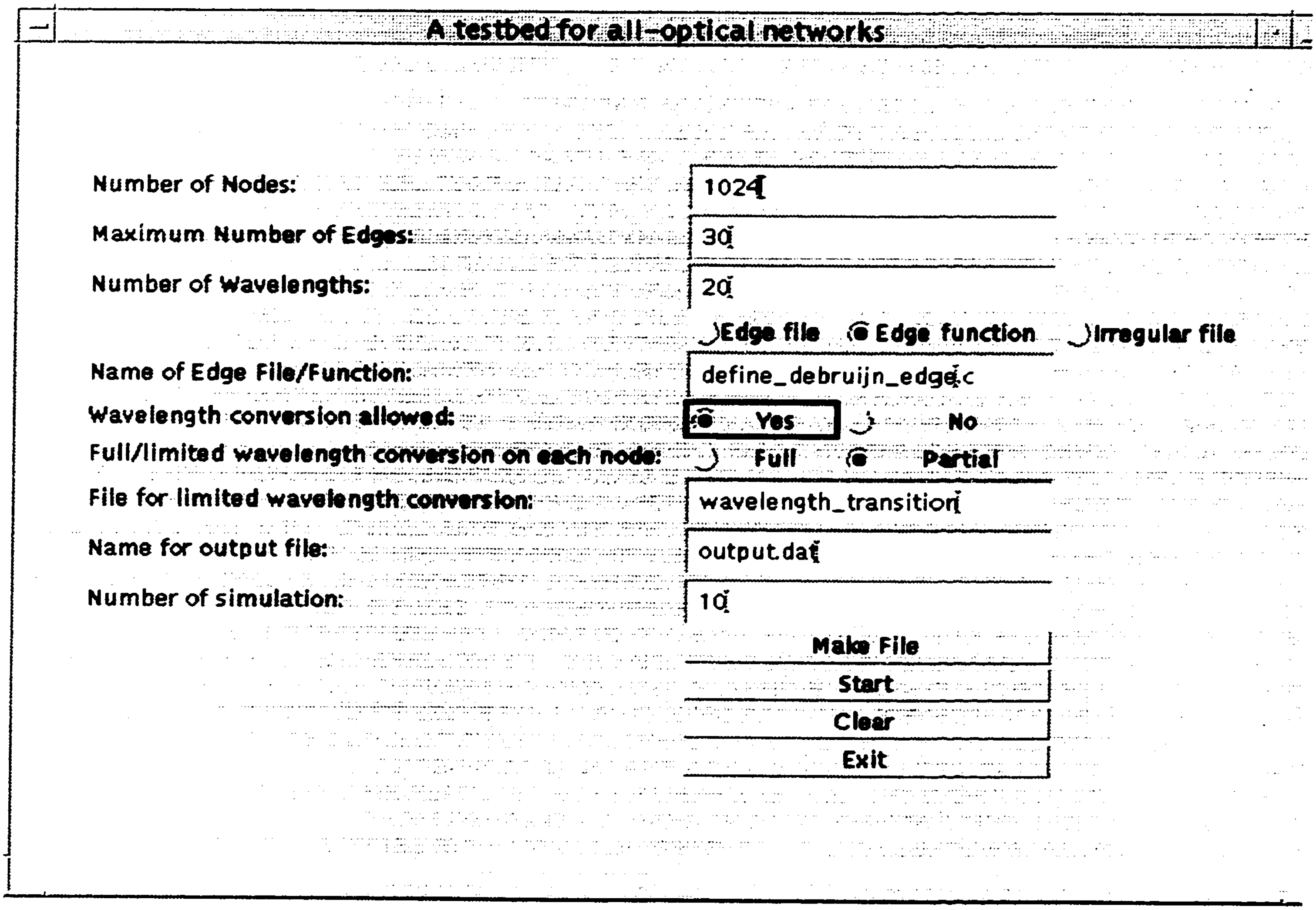
Finally, we run the Java interpreter **java** to start the **Frontend.java**.

As a result, the Java Frontend pops up and waits for users input.

Step 2: Input to user interface

The next step is to type the input for the (4, 5) de Bruijn graph. We follow the exact parameters set for the de Bruijn graph in section 3.3.2. For example, number of nodes for the network is equal to 1024. Fig. 27 shows all the input for the topology.

Figure 27 Input for De Bruijn Graph using the user interface



A testbed for all-optical networks

Number of Nodes: 1024

Maximum Number of Edges: 30

Number of Wavelengths: 20

○ Edge file ● Edge function ○ Irregular file

Name of Edge File/Function: define_debruijn_edge.c

Wavelength conversion allowed: ● Yes ○ No

Full/limited wavelength conversion on each node: ○ Full ● Partial

File for limited wavelength conversion: wavelength_transition

Name for output file: output.dat

Number of simulation: 10

Step 3: Run the simulation

After all input is filled, we first click the “Make File” button in order to create a “make” file to prepare for compilation of all related files. A “make” file serves as a linkage for all files related to **simulation.c**. Once a file is updated in the “make” file, when **simulation.c** runs, the “make” file automatically compiles the file that has been updated. It provides a convenient way to link many different files in the testbed such as routing function **route_debruijn.c** and edge function **define_debruijn_edge.c**.

Here is a typical content for a “make” file:

Content for a “make” file:

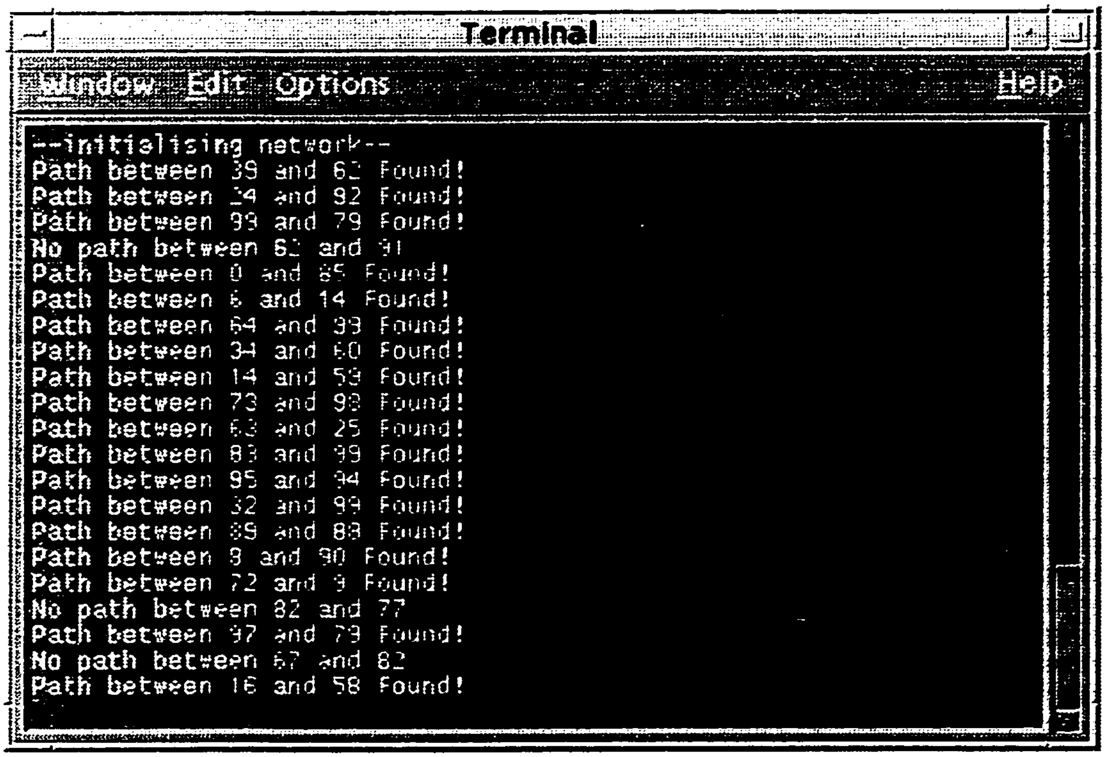
```
simulation: simulation.o define_debruijn_edge.o route_debruijn.o

cc simulation.o define_debruijn_edge.o route_debruijn.o -o simulation

simulation.o: define_debruijn_edge.h route_debruijn.h

define_debruijn_edge.o: define_debruijn_edge.h

route_debruijn.o: route_debruijn.h
```

Next, we press the “Start” button to start the simulation. When the button is clicked, two files are created, namely **input.dat** and **define.h**. The **input.dat** file stores all filenames **simulation.c** needs to refer to such as **wavelength_transition** for information about limited wavelength conversion. On the other hand, the **de-**

fine.h contains all items defined from the input of the user interface such as **#define NUM_NODE 1024** when **simulation.c** needs to be referred to. Afterwards, we invoke the Java native method **NativeMethodImp.c** to run the simulation.

Basically, the Java native method serves as a pipe to send all data to **simulation.c** either in **input.dat** or **define.h**. **NativeMethodImp.c** has two processes. First, it uses a system call **fork()** to create a child process. Second, it compiles the "make" file so that all files related to **simulation.c** are compiled, and runs **simulation.c**. Fig. 28 reveals the process of running the simulation.

Figure 28 Process of running the simulation

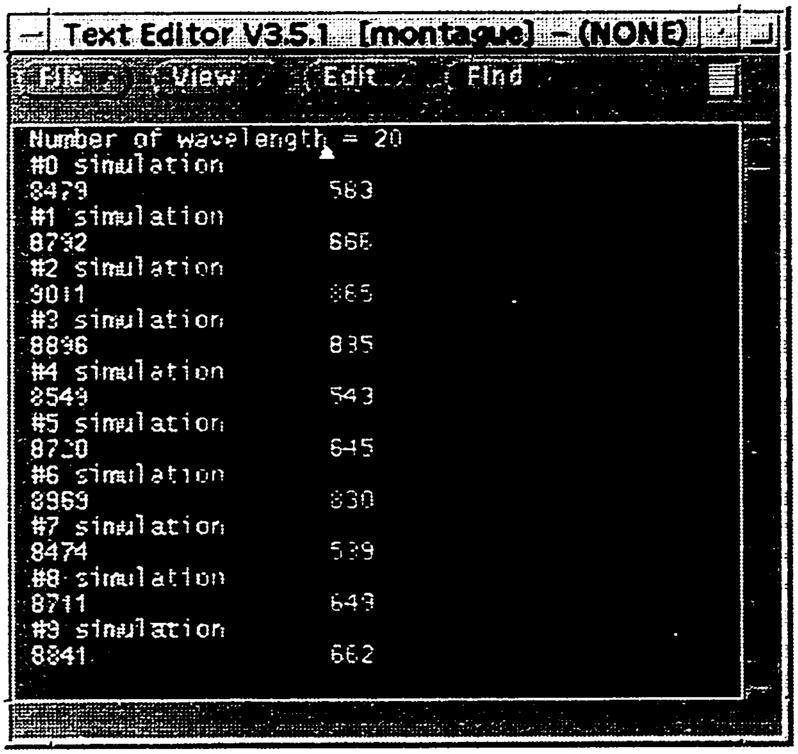


```
--initialising network--
Path between 39 and 62 Found!
Path between 24 and 92 Found!
Path between 99 and 79 Found!
No path between 62 and 91
Path between 0 and 85 Found!
Path between 6 and 14 Found!
Path between 64 and 99 Found!
Path between 34 and 60 Found!
Path between 14 and 59 Found!
Path between 73 and 98 Found!
Path between 63 and 25 Found!
Path between 83 and 99 Found!
Path between 95 and 94 Found!
Path between 32 and 99 Found!
Path between 89 and 88 Found!
Path between 8 and 90 Found!
Path between 72 and 9 Found!
No path between 82 and 77
Path between 97 and 79 Found!
No path between 67 and 82
Path between 16 and 58 Found!
```

Step 4: Get result

After the simulation is done, the testbed creates an output file **output.dat** to store the hits and the misses from the network. Hence we can determine the blocking probability from the output file. Data from **output.dat** is given in Fig. 29.

Figure 29 Output from **output.dat**



```
Number of wavelength = 20
#0 simulation
8479                583
#1 simulation
8732                666
#2 simulation
9011                865
#3 simulation
8896                835
#4 simulation
8549                543
#5 simulation
8720                645
#6 simulation
8969                830
#7 simulation
8474                539
#8 simulation
8711                649
#9 simulation
8841                662
```

BIBLIOGRAPHY

[1] AMP Products & Industries Technology.

[2] S.L. Danielsen, T. Durhuus, B. Mikkelsen, C. Joergensen, and K.E. Stubkjaer. "All-Optical Wavelength Conversion by Semiconductor Optical Amplifiers". *Journal of Lightwave Technology Vol. 14, No. 6*, page 942 — 954, June 1996.

[3] M. Garnot, A. Jourdan, F. Masetti, and M. Sotom. "Wavelength conversion in future WDM optical transport networks". *Digest. IEEE/LEOS 1996 Summer Topical Meetings. Advanced Applications of Lasers in Materials Processing; Broadband Optical Networks - Enabling Technologies and Applications; Smart Pixels; Optical MEMs and their Applications*, pages 47 – 48, 1996.

[4] O. Gerstel. "On The Future of Wavelength Routing Networks". *IEEE Network Magazine*, page 1 — 13, Nov. 1996.

[5] P.S. Henry. "Introduction to Lightwave Transmission". *IEEE Communcation Magazine Vol. 23, No. 5*, page 12 — 16, May 1985.

[6] J.F. Labourdette and A.S. Acampora. "Logically Rearrangeable Multihop Lightwave Networks". *IEEE Transactions on Communications Vol. 39, No. 8*, page 1223 — 1230, Aug. 1991.

[7] K.C. Lee and V.O.K. Li. "A Wavelength-Convertible Optical Network". *IEEE Journal of Lightwave Technology Vol. 11, No. 5/6*, page 962 — 970, May/June 1993.

[8] M.A. Marson, A. Biance, E. Leonardi, and F. Neri. "Topologies for Wavelength-Routing All-Optical Networks". *IEEE/ACM Transaction on Networking Vol. 1, No. 5*, page 534 — 540, Oct. 1993.

[9] B. Mikkelsen, T. Durhuus, C. Joergensen, Pedersen R.J.S. Danielsen, S.L., and K.E. Stubkjaer. "Wavelength conversion devices". *Proceedings, Optical Fiber Communication (OFC '96) San Jose, CA, Vol. 2*, page 121 — 122, 1996.

[10]B. Mukherjee. *Optical Communication Networks*. McGraw-Hill, 1997.

[11]B. Mukherjee. "WDM-based Local Lightwave Networks Part II: Multihop systems". *IEEE Network*, page 20 — 31, July 1992.

[12]B. Mukherjee. "WDM-based Local Lightwave Networks Part I: Single-hop systems". *IEEE Network Vol. 6, No. 3*, page 12 — 27, May 1992.

[13]Deo. Narsingh. *Graph theory with applications to engineering and computer science*. Prentice-Hall, Inc., 1974.

[14]R. Ramaswami. "Multiwavelength Lightwave Networks for Computer Communication". *IEEE Communication Magazine Vol. 13, No. 2*, page 78 — 88, 1993.

[15]R. Ramaswami and A. Segall. "Distributed Network Control for Wavelength Routed Optical Networks". *Proc. of INFOCOM'96. Technical Report RC 20131(89016) IBM Research Report*, 1995.

[16]R. Ramaswami and K.N. Sivarajan. *Optical Networks A Practical Perspective*. Morgan Kaufmann Publishers, Inc., 1998.

[17]R. Ramaswami and K.N. Sivarajan. "Design of Logical Topologies for Wavelength-Routed Optical Networks". *IEEE JSAC/JLT Special Issue on Optical Networks*, June 1996.

[18]R. Ramaswami and K.N. Sivarajan. "Routing and Wavelength Assignment in All-Optical Networks". *IEEE/ACM Transaction on Networking*, page 489 — 500, Oct. 1995.

[19]A. Senguta, S. Bandyopadhyay, and A. Jaekel. "On the performance of dynamic routing strategies for all-optical networks". *presented at the IASTED Conference on Parallel and Distributed Computing and Networks, Singapore,*, Aug. 1997.

[20]K.N. Sivarajan and R. Ramaswami. "Lightwave Networks Based on de Bruijn Graphs". *IEEE/ACM Transactions on Networking Vol. 2, No. 1*, page 70 — 79, Feb. 1994.

[21]M.E. Steenstrup. *Routing in Communications Networks*. Prentice Hall, 1995.

[22]S. Subramaniam. *All-optical networks with sparse wavelength conversion*. PhD thesis, University of Washington Seattle, 1997.

[23]J. Yates, J. Lacey, D. Everitt, and M. Summerfield. "Limited-Range Wavelength Translation in All-Optical Networks". *Proc. INFOCOM'96*, page 954 — 961, Mar. 1996.

[24]S.J.B. Yoo. "Wavelength Conversion Technologies for WDM Network Applications". *Journal of Lightwave Technology Vol. 14, No. 6*, page 955 — 966, June 1996.

[25]H. Zeineddine. Effect of Limted Wavelength Conversion in All-Optical Networks Based on de Bruijn Graphs. Master's thesis, University of Windsor, 1997.

[26]Z. Zhang and A.S. Acampora. "A Heuristic Wavelength Assignment Algorithm for Multihop WDM Networks with Wavelength Routing and Wavelength Re-use". *IEEE/ACM Transactions on Networking Vol. 3, No. 3*, page 281 — 288, June 1995.

VITA AUCTORIS

Tony Lam was born in 1972 in **Hong Kong**. He graduated from **Templeton Secondary School** in 1992 in Vancouver. From there he went on to the **University of Windsor** where he obtained a B. Sc. in Computer Science in 1996. He is currently a candidate for the Master's degree in Computer Science at the University of Windsor and hopes to graduate in the Spring of 1999.