ABSTRACT

IYER, VIJAY R. A Simulation Study of Wavelength Assignment and Reservation Policies with Signaling Delays. (Under the direction of Professor George N. Rouskas).

This thesis studies the effect of non-negligible signaling delays on the performance of wavelength-assignment heuristics, wavelength reservation schemes, routing schemes, holding time (average being $1/\mu$) of the lightpaths and traffic loads (average being λ/μ), in second-generation optical wide area networks (WANs). A network simulator was developed using the C++ language for this study. The simulator supports any input topology with single or multi-fiber links, many routing schemes (static, alternate and dynamic), dynamic traffic loads, and may be modified easily to accomodate different wavelength-assignment policies. The signaling messages used, in our study, to establish lightpaths, follow the Constrained-Routing Label Distribution Protocol (CR-LDP) semantics. The problem studied here falls under the general category of Routing and Wavelength Assignment (RWA) Problem which has been proved to be NP-hard. Previous studies have mostly considered static routing (with static or dynamic traffic demand), and static traffic demand (with static or alternate routing) under zero propagation delays. A few papers in the recent past have studied the effect of signaling delays but have been limited in scope. We study the effect of varying holding times, compare random versus first-fit wavelength assignment policy, compare fixed versus alternate routing, compare backward wavelength reservation schemes to forward reservation schemes, and lastly study the effect of traffic loads. We find that, in general, the random wavelength assignment policy performs better than first-fit policy and that under certain conditions, alternate routing scheme performs worse than fixed routing scheme.

A Simulation Study of Wavelength Assignment and Reservation Policies with Signaling Delays

by

Vijay R. Iyer

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Master of Science in Computer Networking

Department of Computer Science

Raleigh

2002

Approved By:

Dr. Douglas S. Reeves

Dr. Rudra Dutta

Dr. George N. Rouskas
Chair of Advisory Committee

To my parents

T G Ramachandran

Ambuja Ramachandran

for having taught me the importance of a good education early on in my life

and my sister

Uma Iyer

for her unconditional love.

Biography

Vijay Iyer was born in a small town called Dombivili, near the city of Mumbai, India. After finishing his high school in Mumbai, he graduated with a Bachelor of Technology ($B.Tech$) degree in Chemical Engineering from the Indian Instiute of Technology (IIT)-Bombay. He worked for a year as a Management Trainee in Britannia Industries Ltd. He then worked for about 6 months as an Analyst in a startup called SNR India (now Cash-Tech) Pvt.Ltd. At the time of this writing, he is working towards his Master of Science (MS) in Computer Networking in North Carolina State University (NCSU) in Raleigh and actively looking for a job.

Acknowledgements

I would like to acknowledge the efforts of my advisor, Dr.Rouskas, for giving me direction and guidance when required. He helped me focus and I had the pleasure of working with him.

I would like to acknowledge Dr.Reeves and Dr.Dutta for agreeing to be on my thesis committee.

Among my colleagues, I would like to thank Prashant Gupta with whom I have had many a discussion on designing a 'good' simulator, Prashant Iyer who helped me spot C++ errors when I could not (thereby reducing time spent on debugging), Akshay Adhikari who accelerated my learning curve w.r.t gnuplot package, and Lina Battestilli who helped me learn Latex faster by providing me with a template file.

The last two and half years have been more than a mere academic stint for me. I would like to express my gratitude to Girish and Prabhas for having been such good roomies and friends, Sandeep for all the entertainment he provided, PeeGee for having introduced me to 'trance' music and with whom I have had many serious and frivolous conversations, Akshay for reminding me what it takes to be a sane person, Lakshmi for being insane yet responsible, crazy yet loveable, and James Jong for giving me a patient ear whenever I got philosophical. Last, but definitely not the least, I would like to express my sincere thanks to my very good friend and favorite project-mate, Lina Battestili, for ... everything! It was just amazing the way we gelled despite our completely different cultural backgrounds.

Contents

List of Figures

Chapter 1

Introduction

The demand for increased bandwidth at lower costs is driving the growth of new Internet technologies. Applications such as the world wide web (WWW) that involve transfers of data, voice clips and video images use up the bandwidth which in turn causes investments to increase the bandwidth made available to consumers. Newer applications making use of this increased bandwidth came up and this cycle was going on until the recent telecom bust. The estimates for traffic demand in the future however, show no signs of decrease. To satisfy the enormous demand in bandwidth, enormous capacity is needed. Optical networking is one such technology that addresses this problem very well. The development of optical networks was greatly driven by the Telecom Deregulation Act which led to many new network service providers hoping to build their networks with the latest technology. Optical Networks may be classified as:

1. *First Generation Optical Networks:* These optical networks involved replacing copper cables by optical fibers as the medium of transmission as optical fibers had better physical characteristics for transmission requirements than copper cables. The raw bandwidth offered by optical fibers is huge. Hence the copper cables in core telecom networks, which face loads more than perhaps any other networks, were first replaced by optical fibers. The switching and processing of bits were, however, handled in the electronic domain as before. Optical fibers were preferred for bit rates greater than 10 Mbps that were needed to be transmitted over a distance of more than a kilometer.

Examples of first generation optical networks are SONET/SDH networks that form the core telecom networks in North America, Europe and Asia. Other examples include the FDDI-based enterprise networks. From a network layering point of view, the impact of the first generation optical networks was felt primarily in the *physical layer*. From hereon, there were primarily two fundamental ways of increasing the speeds in the networks; either increase the electronic processing speeds by improved time division multiplexing (TDM) techniques or increase the capacity by using multiple carrier wavelengths in the fiber at the same time. The latter technique is similar to frequency division multiplexing (FDM) techniques in radio systems and offers the flexibility of working at the maximum state-of-the-art electronic speeds.

2. *Second Generation Optical Networks:* These networks were made capable to use multiple carrier wavelengths that were multiplexed onto a single fiber thus offering increased bandwidth. The technique is called *Wavelength Divison Multiplexing (WDM)*. The primary improvement of second generation optical networks over their first generation counterparts from technological point of view was in incorporating the switching and routing functionality in the optical domain and allowing for transparency of data format, protocol and bit rates. It thus allowed for lesser electronic load on a node by ensuring the need to terminate the traffic intended only for that node while allowing the other traffic to cut right through the node in the optical domain. In the first generation networks, a node would have to terminate all the optical signals (irrespective of whether they are intended for itself or not), convert them to electronic signals, process them and then regenerate the traffic not intended for itself into optical signals and send them on the appropriate outgoing links. The second generation optical switches are called the *Optical Cross-connects (OXCs)*. These switches may be configured to switch the optical signals between any incoming port to any outgoing port. Second generation networks were designed in a way so that the optical layer offers a *lightpath service* to the layers above.

   Lightpath service: Lightpaths are connections (circuits) that are set up between two nodes. They are established by setting up a wavelength along each link in the path from the source node to the destination node. However, the chosen wavelength need not be the same in all links. It depends on the wavelength conversion capability of an intermediate node along the path. When none of the nodes in the network have that capability, the constraint of having to choose the same wavelength along all the links in the path is referred to as the *wavelength continuity constraint*. The granularity of bandwidth made available by a lightpath service to the layers above is in terms of a full wavelength. One wavelength can offer bandwidth to the order of a few Gbps.

1.1 Focus and Contribution

The work presented in this thesis is based on a simulation study. It takes into account the delay encountered by the signaling messages combined with various routing schemes, wavelength assignment policies and wavelength reservation schemes that are used to establish lightpaths in second-generation optical WANs. The performance measure of interest used is blocking probability. To the best of our knowledge and belief, such work of combined nature has not been done so far. Our results include the following:

- As the holding time of a lightpath relative to the signaling delay increases, the blocking probability decreases for all hop lengths, other factors kept constant.
- Random wavelength assignment policy performs mostly better than first-fit assignment policy.
- Alternate routing scheme need not always perform better than fixed routing scheme.
- Between the two wavelength reservation schemes studied, the backward reservation scheme performs better than the forward one.
- As traffic load increases, the blocking probability increases.

The main component of the signaling delay is the propagation delay. Processing delay and queueing delay are assumed not to vary much and hence considered as factored into the overall delay. The signaling messages simulated in this study follow the semantics of *Constrained-Routing Label Distribution Protocol (CR-LDP)* [9], a relatively new signaling protocol proposed by the Multiprotocol Label Switching (mpls) working group in Internet Engineering Task Force (IETF). The topologies studied include 12-node ring, 5×5-torus and NSFNET topologies. The traffic demand is not known prior to the simulation and is generated dynamically. More information about the simulation parameters used and how they affect the establishment of lightpaths may be found in Chapter 2.

1.2 Organization of Thesis

The rest of the thesis is organized as follows. Chapter 2 discusses the optical network components and their architectures. Chapter 3 explains the problem of establishing lightpaths and contains some literature review. It includes a brief description of *Multi-Protocol Label Switching (MPLS)*, *Label Distribution Protocol (LDP)* and CR-LDP. Chapter 4 explains, at a high-level, the modules of the simulator that has been built for this thesis. Chapter 5 discusses the results and trends. Chapter 6 concludes the thesis and discusses the future work.

Chapter 2

Optical Network Architecture

It would be useful, as a first step, to briefly describe the components and the architecture of optical networks before describing the modeling of such a network and listing the system parameters considered in this thesis.

2.1 Optical Network Components and Architecture

In this section, we would like to mention some of the optical components and how they enable in supporting architectures that tend towards all-optical networks.

Point-to-point WDM Systems: A block diagram of a point-to-point WDM system is shown in Figure 2.1. The diagram illustrates two nodes connected by a single fiber link (assume it is a directed link for simplicity). The components involved in optical networks are:

1. *Terminal equipment (TE):* It terminates the optical signal and converts it into electronic signals so that they maybe processed at the node. Each TE is connected to a WDM transmitter that transmits a single (and typically, a fixed) wavelength.

2. *Multiplexer/Demultiplexer (Mux-Demux):* It is placed at the transmitting and receiving ends of the fiber link. The multiplexer helps in combining the various wavelengths coming from the different TE's, modulates them and put them on the fiber. The

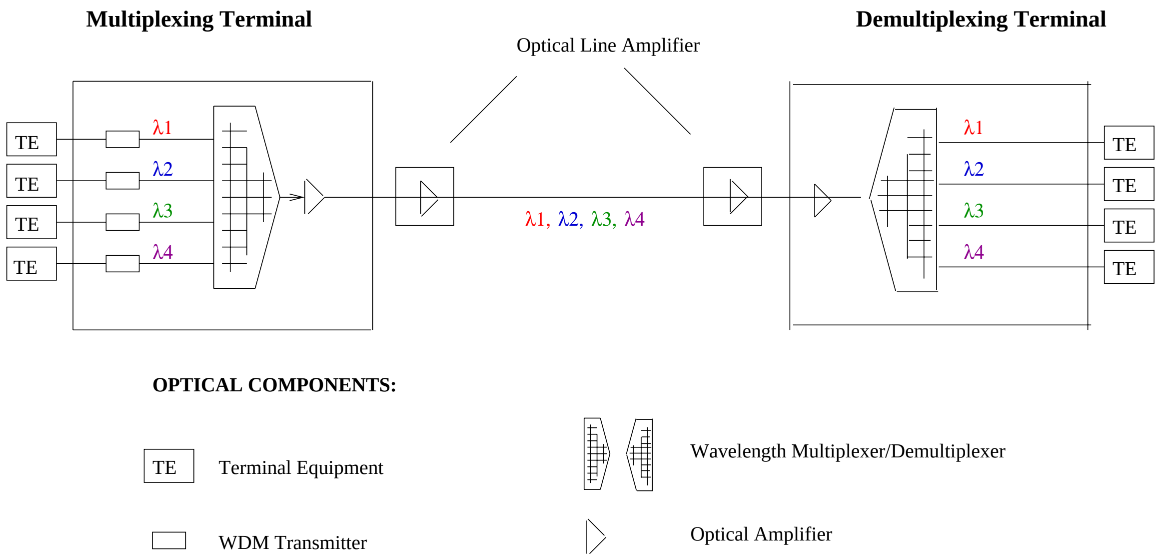

Figure 2.1: Point to Point WDM Systems [19]

demultiplexer, at the receiving end, helps separate out the wavelengths that were combined due to the multiplexing.

3. *Optical Amplifiers:* Along the fiber link, there could be optical amplifiers that could help perform a subset or all of regenerating (boosting only the power of the signal), reshaping (producing original pulse shape of the bit), and retiming (helping the signal synchronize to its original bit rate) the optical signal to overcome the harmful effects caused due to the phenomenon of attenuation, dispersion, and other non-linearities. When done together the amplification is called as 3R. All-optical amplification performs only 1R (regeneration of signal) and differs from optoelectronic amplification that performs either 2R (regeneration and reshaping) or 3R [5]. The amplifiers that were developed initially mandated the need to convert the optical signals to electronic signals in order to perform the 3R operations, and then transmit them back as optical signals. The 2R and 3R techniques are thus clearly less transparent than the 1R technique. With the aggregate bit rate of a few channels the 2R and 3R operations may not be practical and hence 1R amplification is the choice of future all-optical networks.

   Notice that all wavelengths are necessarily converted to electronic signals (some of the wavelengths might be intended for the current node and hence not re-transmitted)

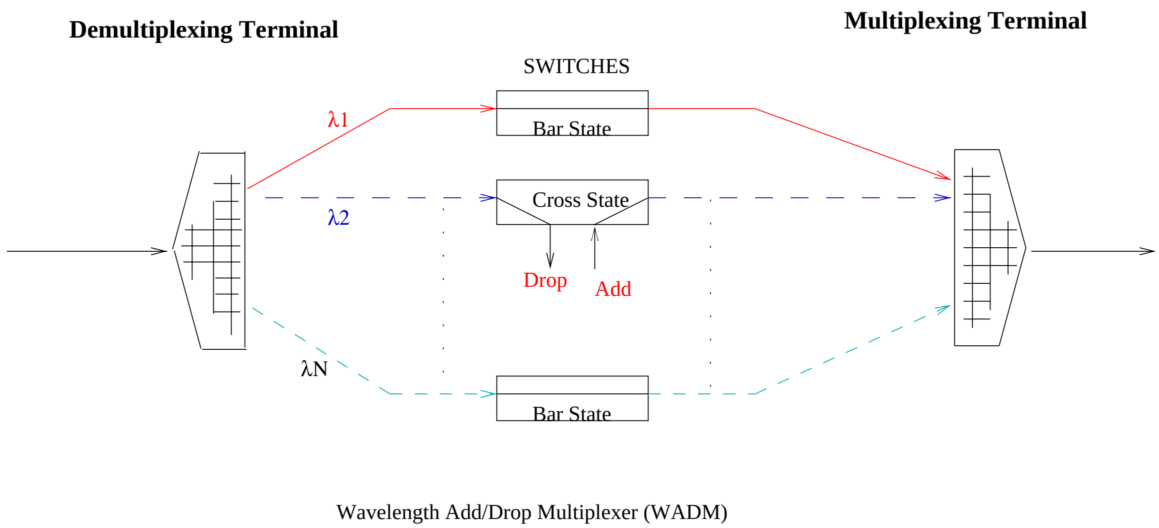

Figure 2.2: Wavelength Add/Drop Multiplexer (WADM) [19]

at the node irrespective of whether the information carried on these wavelengths is intended for that node. Thus, if wavelengths were allowed to cut through the node without mandatory termination at each node, it would enhance the performance of the information transfer through the network. This is facilitated by *Wavelength Add/Drop Multiplexers (WADM).* They are also called as *Optical Add/Drop Multiplexer (OADM)* sometimes. A WADM can be 'inserted' on a physical fiber link. A WADM is briefly described below.

4. *Wavelength Add/Drop Multiplexers (WADM):* A typical WADM is shown in Figure 2.2

   It consists of a demultiplexer that separates the wavelengths. Each separated wavelength is then sent via a 2×2 switch. The state of the 2×2 switch (bar or cross) decides the fate of the wavelength. Typically, the bar state allows a wavelength to continue uninterrupted thus facilitating the wavelength to cut through the node without being terminated or converted to electronic form for processing. The switch can be configured electronically to the cross-state and this causes the signal on the corresponding wavelength to be 'dropped' (terminated) locally at the node. A new data stream on the same wavelength may now be 'added' on the outgoing link. More than one wavelength maybe dropped and added at a node if the node is equipped with the necessary hardware.

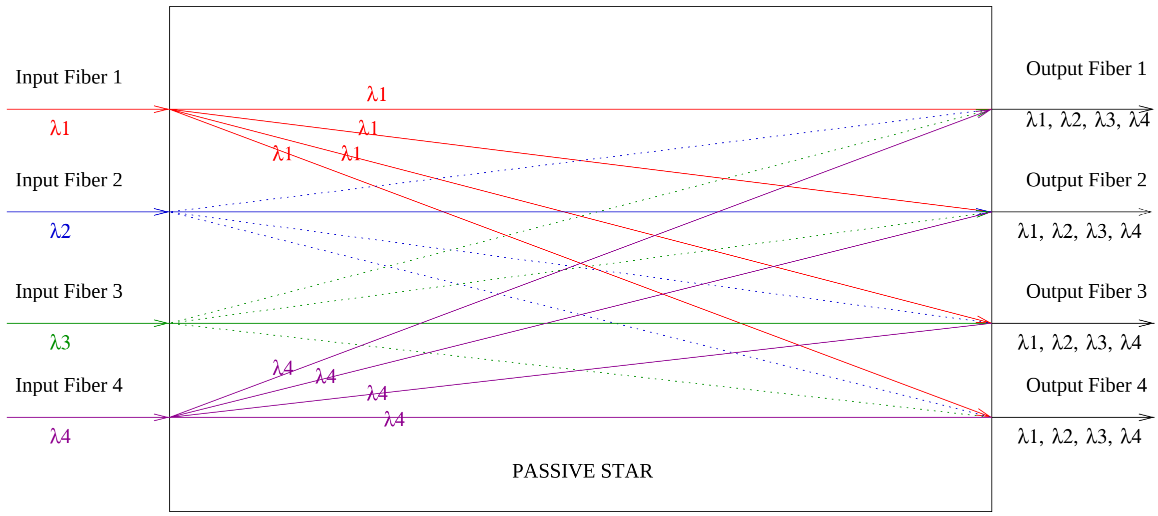

Figure 2.3: Passive Star [19]

Network of multi-wavelength links: We have seen above the components needed for a point-to-point WDM system and how a wavelength may be allowed to cut through a node without the need of terminating at every node which is not the destination node. The question that comes to mind next is how a network of multi-wavelength fiber links may be achieved and what key components are needed for the same. These components can be broadly called as *fiber interconnection devices*. They fall under three categories:

1. *Passive Star:* The block diagram for a passive star is shown in Figure 2.3. A passive star is essentially a *broadcast* device, so that the power of an input signal is split across all the outgoing links equally and appears as the same wavelength at all outgoing ports. Thus the number of total simultaneous connections that may be supported by a passive star is restricted and is equal to the minimum of the two numbers viz. number of wavelengths and the number of fiber ports. Passive stars are used to build local WDM networks (WDM-based LANs) that generally have a star topology. Optical WANs do not exist in star topology and hence star topologies are not of immediate interest to us.

2. *Passive router:* The block diagram for a passive router is shown in Figure 2.4. A passive router is capable of routing a wavelength on any input port to a pre-determined output port on the same wavelength. This capability of the passive router allows for

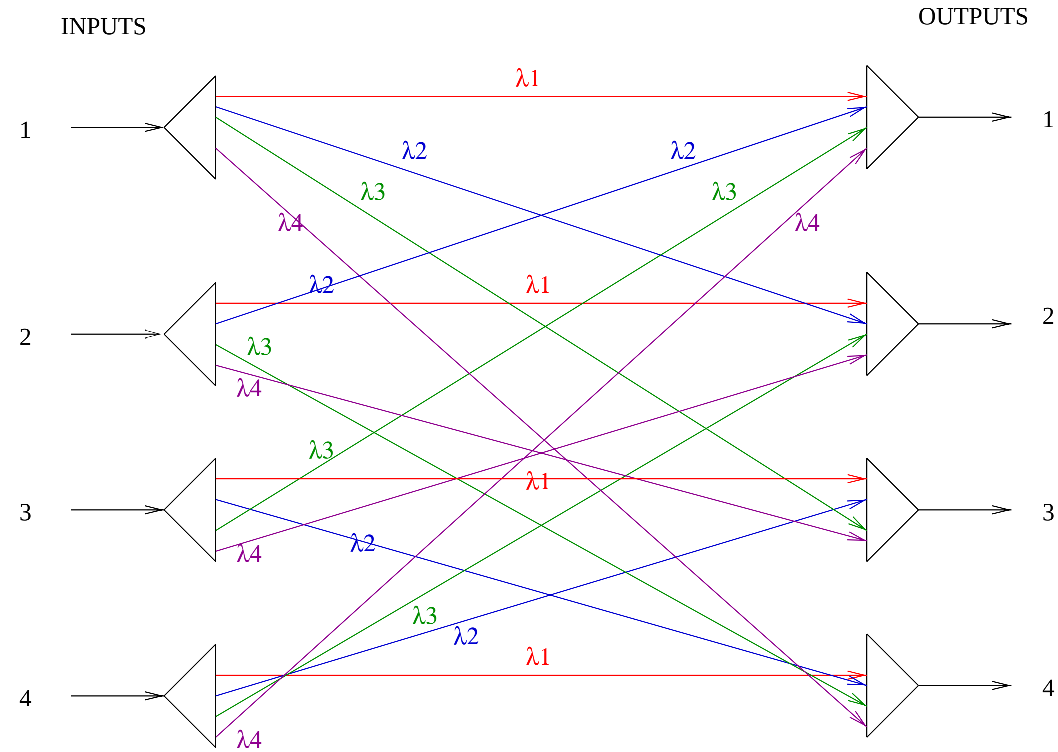

Figure 2.4: Passive Router [19]

spatial re-use of the wavelengths within the passive router (as opposed to passive star where only one instance of a certain wavelength, say the blue wavelength, could be present in the star at a given time). Thus, given N total wavelengths, and P input and output ports, the number of simultaneous connections that could be supported by a passive router is NP, as opposed to just $\min\{N, P\}$ by passive star. The pre-determined routing of a wavelength on a certain input port to a specified output port and wavelength is determined by the "routing matrix" that characterizes the passive router. The routing matrix is fixed ("hard-wired") and cannot be changed. Clearly the flip side in using the passive router is that it lacks the native support for broadcast. Thus it is primarily a *unicast* device and is mainly used as a mux-demux device in optical networks.

3. *Active Switch (or Optical Cross Connect, OXC):* The block diagram for an active switch is shown in Figure 2.5.

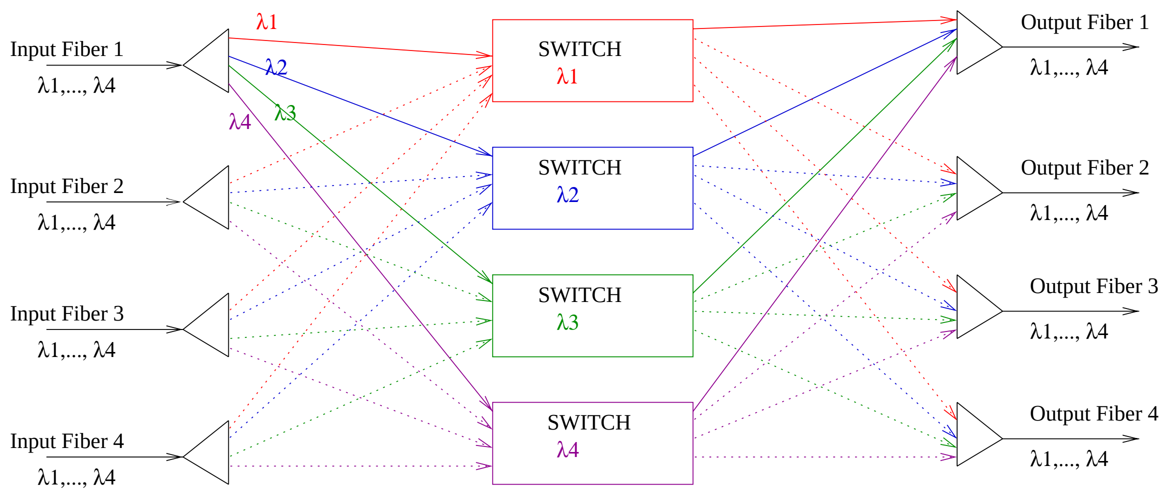

Figure 2.5: Active Switch (OXC) [19]

An active switch has one important property that enhances its value in providing connections to higher layers. Its routing matrix may be reconfigured electronically on demand thus allowing for routing a wavelength incident on a certain input port to any of the output ports. An active switch is sometimes called the *Wavelength-Routing Switch (WRS)*, or the *Photonic-Routing Switch (PRS)*, or simply the *Optical Cross-connect (OXC)*. It can be also be provided with a wavelength conversion facility that can convert the wavelength on the input port to a different wavelength which could then be routed to any of the output ports. This is implemented before the mux stage of the output fiber. When the wavelength conversion facility is provided to the OXC, it is called *Wavelength Interchanging Cross-connect (WIXC)* or referred to as an OXC with wavelength-conversion capability. The flip side to such flexibility provided by the active switch is that it must always be "powered up" and is not as fault tolerant as the passive star and passive router are. Also, each OXC has as many internal switches as the number of wavelengths it needs to support. Each internal switch handles a specific wavelength from all input ports. This structure adds to the cost of the OXC equipment as opposed to its passive counterpart. OXCs are used for building optical WANs.

Thus using the above components, a high-level view of the wavelength-routed optical network would look like as shown in Figure 2.6. Each node represents an active switch ca-

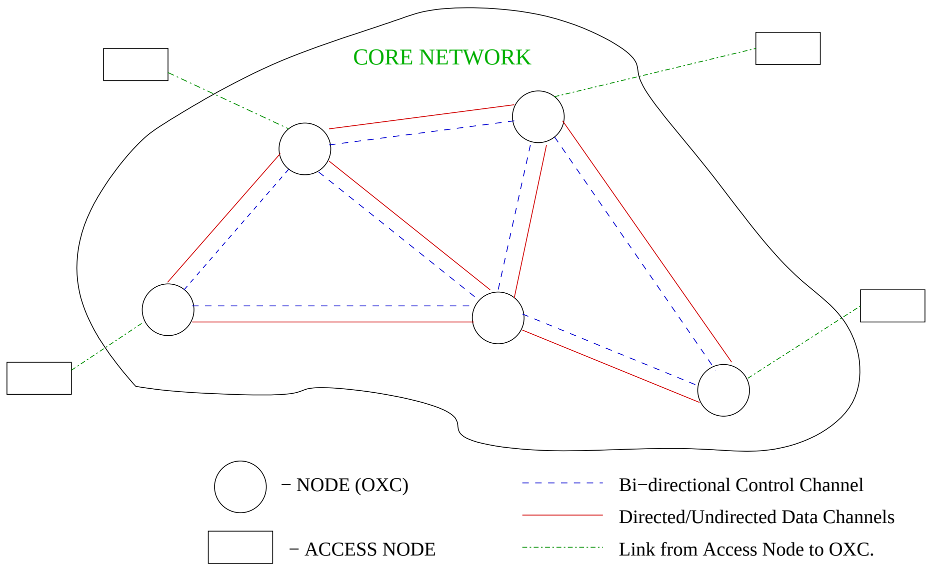

Figure 2.6: Wavelength Routed Network [19]

pable of routing wavelengths. The links are shown to be single-fiber links here, though in practice they are usually multi-fiber links. In optical MANs and WANs, the OXCs are inter-connected using fiber-links and this inter-connection forms the physical topology of an optical network.

2.2 System Parameters

We are now in a position to list and describe the system parameters used for modeling a network such as described in Figure 2.6. Given a physical topology (manually input or randomly generated) the system parameters are:

1. **Traffic information:** The traffic matrix may be known apriori (static traffic demand) or may not be known (dynamic traffic demand). In both traffic demand types, there are various options of traffic distribution (Poisson, Normal, Uniform etc.) to choose from. Since establishing lightpaths is generally modeled as a circuit-switched problem (in terms of arrival request and holding times), Poisson distribution is generally accepted as a suitable and valid traffic distribution, and is often used. Traffic load is generally represented by the ratio of arrival rate (λ) to the service rate (μ).

This ratio may be specified for the entire network or for each source-destination pair or for a group of source-destination pairs. The service rate (μ) could be chosen as a factor of the diameter of the physical topology, which is computed in terms of either maximum number of hops or maximum round trip time (RTT) that exist between a source-destination pair in the graph.

2. **Routing scheme:** Fundamentally there are two ways of routing, *source routing* and *hop-by-hop* routing. Source routing refers to the computation of the route right at the source and the traffic follow only that path. (The terms path and route may be used here interchangeably). For this scheme to succeed, the source node (and hence all nodes, as they are equally likely to be the source node) is expected to have a good idea of the current state of the network. It works well in cases where the network is stable and thus not too much change, such as failure of nodes and links, is expected. Hop-by-hop routing consists of the packet finding its way to the destination based on the information obtained from the current (visiting) node. Thus the onus is on the visiting node to provide some information in order to direct the incoming packet. Again, all nodes are required to have as consistent a view of the network as possible. Since the path is 'discovered' by the packet as it moves along, and not computed at the source, it is possible that two packets that have the same source-destination pair may each find a completely different path to the destination provided they are not subjected to a connection-based approach (example: IP routing). Source routing (CR-LDP semantics) is employed in this study. Within this scheme, there are two flavors of routing viz. *Static Routing* and *Dynamic Routing*. Dijkstra's shortest algorithm is typically employed to determine the shortest path in both cases. Static routing means that the shortest path for a source-destination pair is pre-computed based on a certain metric (hop count, for example) and is used as the path along which the signaling messages for establishing a lightpath having the same source-destination pair is to be routed. The number of re-attempts (in case the first attempt fails) at setting up the lightpaths maybe set to more than one. Each re-attempt then uses a link-disjoint pre-computed path for a given source-destination pair.

   Dynamic routing, however, involves, computing the shortest paths on the fly. In this scenario, the shortest paths between any pair of nodes need not be the same anymore. The "link-cost" functions, based on the availability of wavelengths for example, may be used to determine the weights of the link. Since Dijkstra's algorithm computes shortest paths based on weights of links in a graph, the shortest paths for a given pair of nodes need not be the same every time in a dynamic routing scheme as the weights of links change with time. The number of re-attempts are generally set to one in dynamic routing schemes as the shortest path between a source-destination pair is not expected to change (unless discovery of link-disjoint paths is enforced every time a new path is computed) while computing immediately after the first attempt fails, especially in zero-delay scenarios.

3. **Number of wavelengths per link (W):** Each directed link will be capable of supporting W wavelengths at any given time and this information is an important system parameter that affects the performance of the network.

4. **Wavelength Reservation Scheme:** There are fundamentally two ways of reserving wavelengths. The first one reserves all the available wavelengths in the links and carries this information as the signaling messages propagate forward along the route to the destination. This is called the *Forward Reservation Protocol (FRP)*. It is shown in Figure 2.7. Upon reaching the destination, the destination chooses one of the several reserved wavelengths according to a *wavelength assignment policy* (such as first-fit, random etc.; more on this later in this section) and the signaling message then releases the wavelengths other than the chosen one while traversing back to the source node. The second way of reserving wavelengths probes the availability of wavelengths (but does not actually reserve them) and collects that information as the signaling messages travel forward on their way to the destination node. Once the signaling message reaches the destination node, the destination node selects one of the many 'available' wavelengths according to a wavelength assignment policy. The signaling message then attempts to confirm this selected wavelength along each link in the path while traversing in the reverse direction. This is called the *Backward Reservation Protocol (BRP)*. It is shown in Figure 2.8. The signaling messages in a BRP scheme do not need to release any wavelengths on their return to the source node (as in FRP scheme) for they do not hold up any resources while traversing in the forward direction.

   There are various flavors of both FRP and BRP schemes. Clearly the scheme of reserving all avaiable wavelengths in FRP scheme or adding to the choice-list (but

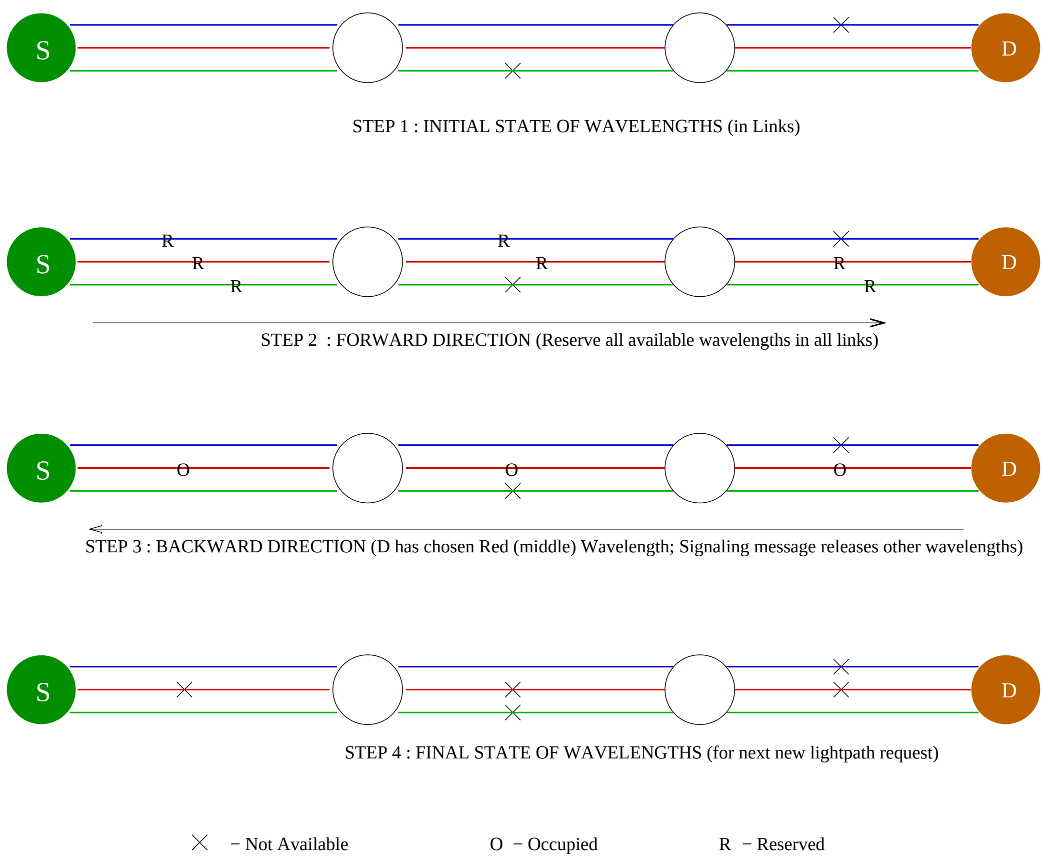

Figure 2.7: Forward Reservation Protocol(FRP) - Normal Scheme

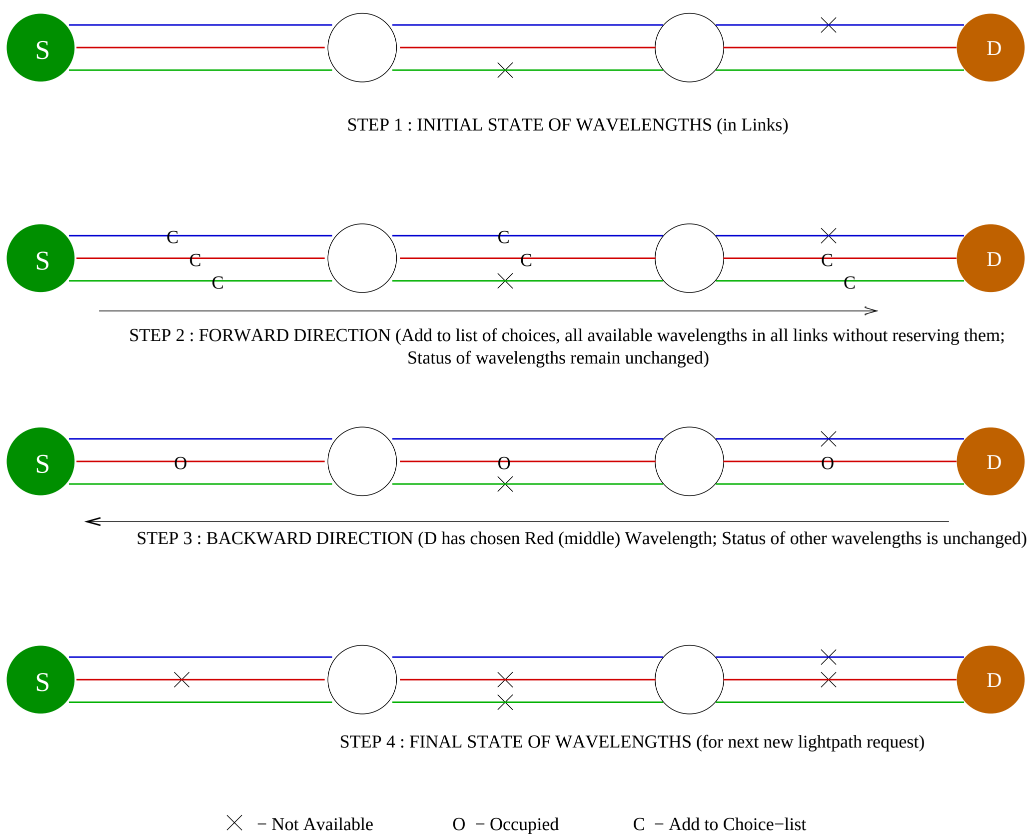

Figure 2.8: Backward Reservation Protocol(BRP) - Normal Scheme

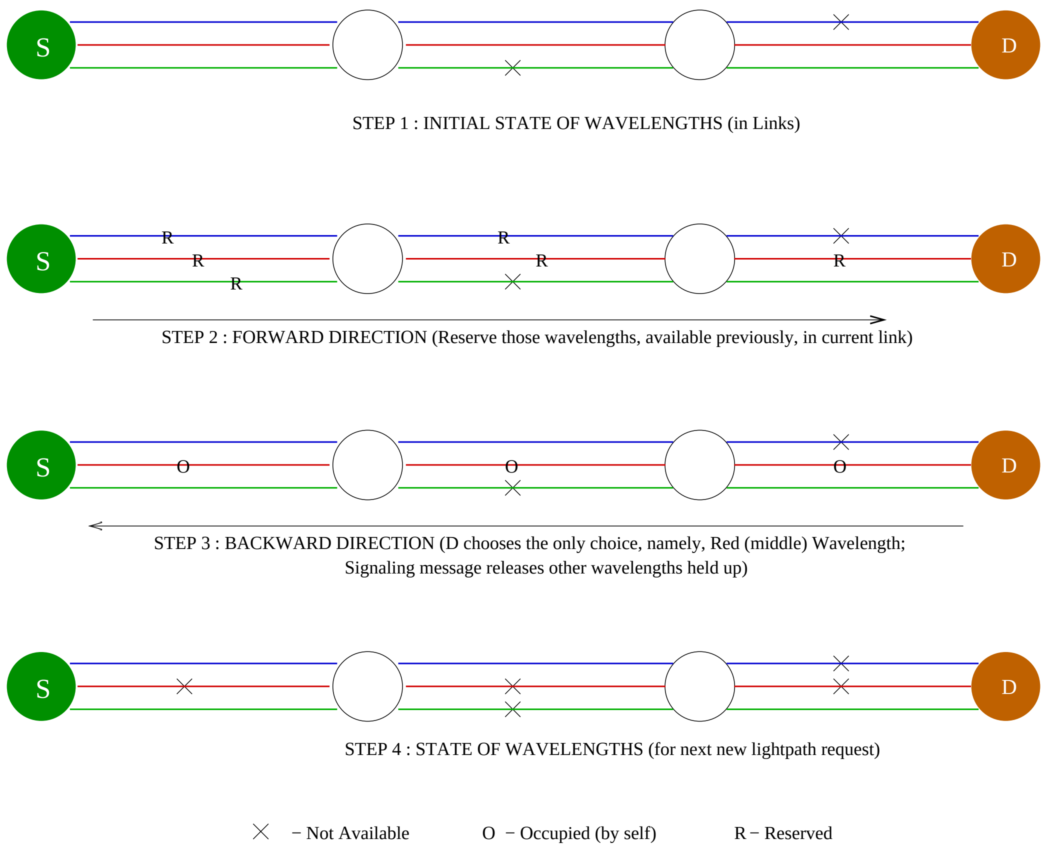

Figure 2.9: Forward Reservation Protocol(FRP) - Subset Version

not reserving) in BRP scheme, as illustrated in Figure 2.7 and Figure 2.8, in a link for one lightpath request would lead to decreased performance of the network because the wavelengths reserved in one link may not be available in another link; yet their being reserved in FRP scheme (or selected in BRP scheme) leads to no usefulness for the establishment of the lightpath. In fact this scheme harms the chances of other lightpaths in FRP scheme. As a result more blocking results. An acceptable variant ([18]) of this protocol is that the signaling messages reserve (or add to choice-list) only those available wavelengths that have been reserved till the previous hop. The diagrams shown in Figure 2.9 and Figure 2.10 illustrate this. We call this scheme as *FRP-SS (FRP Subset)* and *BRP-SS (BRP Subset)* and will be the flavor of FRP and BRP schemes that would be studied here.

Clearly many lightpaths competing for the same resources simultaneously may lead to

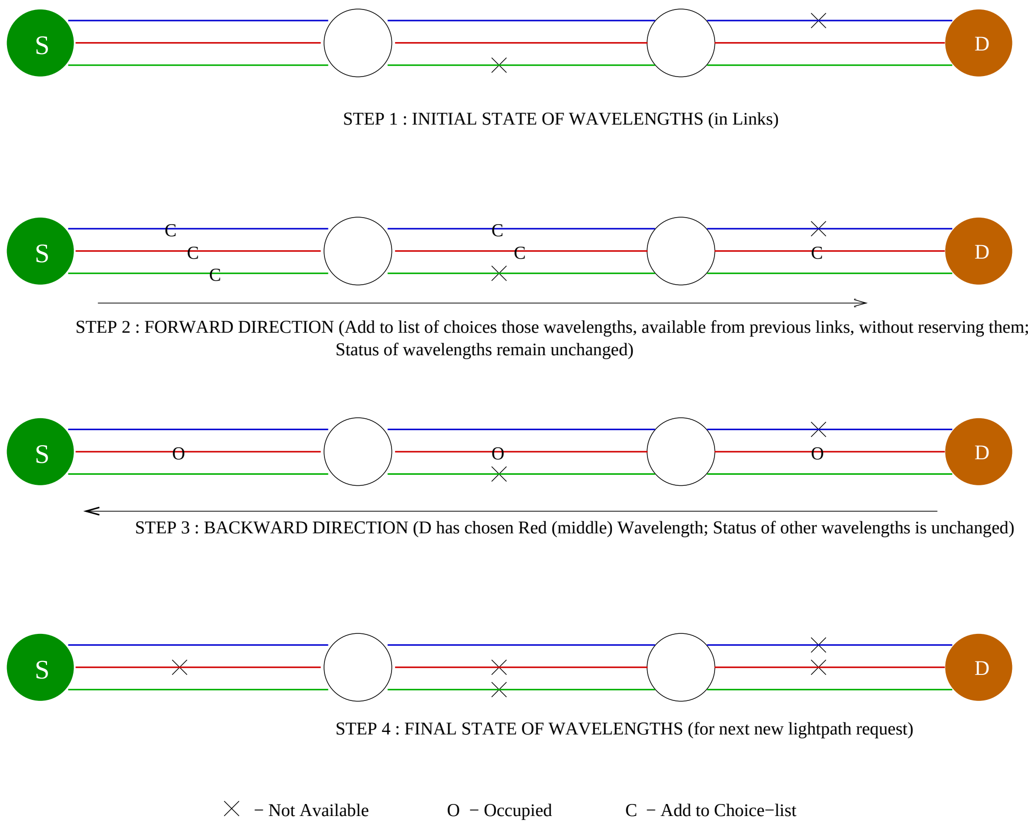

Figure 2.10: Backward Reservation Protocol(BRP) - Subset Version

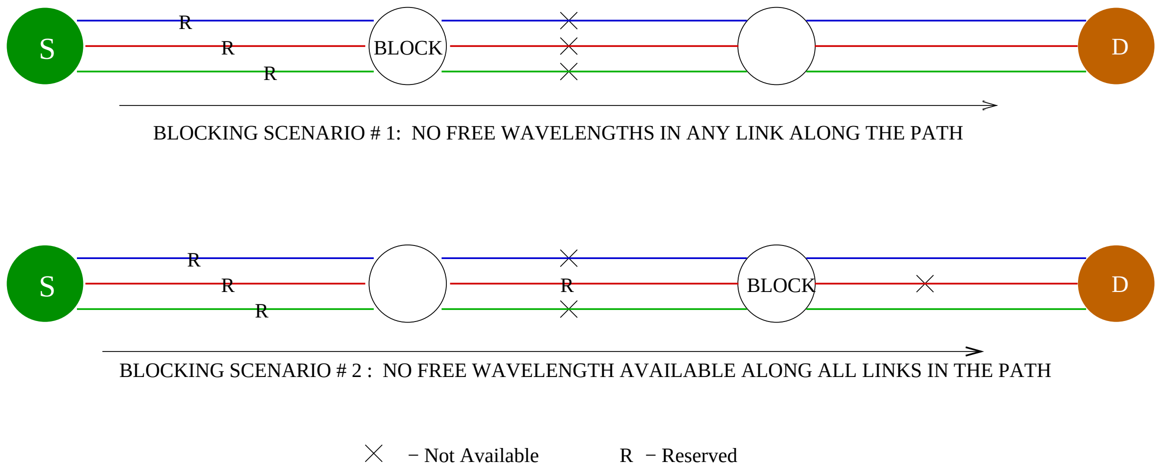

Figure 2.11: Blocking in FRP - Subset Version

blocking (unavailability of resources) of one or more of the lightpaths. The reasons due to which blocking might occur differ depending on the type of wavelength reservation scheme employed.

Blocking in FRP-SS: In this scheme, a lightpath request could get blocked due to the non-availability of a single available wavelength in the link (that is part of the source routed path) along which the signaling message is supposed to travel. The non-availability of the wavelength may be due to two reasons.

- All the wavelengths are either in-use or reserved (but not in use) by other lightpaths established along that link.
- Inability to satisfy the wavelength continuity constraint due to the current state of reservation, or the wavelengths are used up by other lightpaths along the links of our interest.

The diagram in Figure 2.11 illustrates the above points.

Blocking in BRP-SS: The lightpath request could get blocked due to three reasons.

- All the wavelengths are used up for lightpaths established along that link. Thus this information would get captured in the probing phase (forward direction of signaling messages) itself.

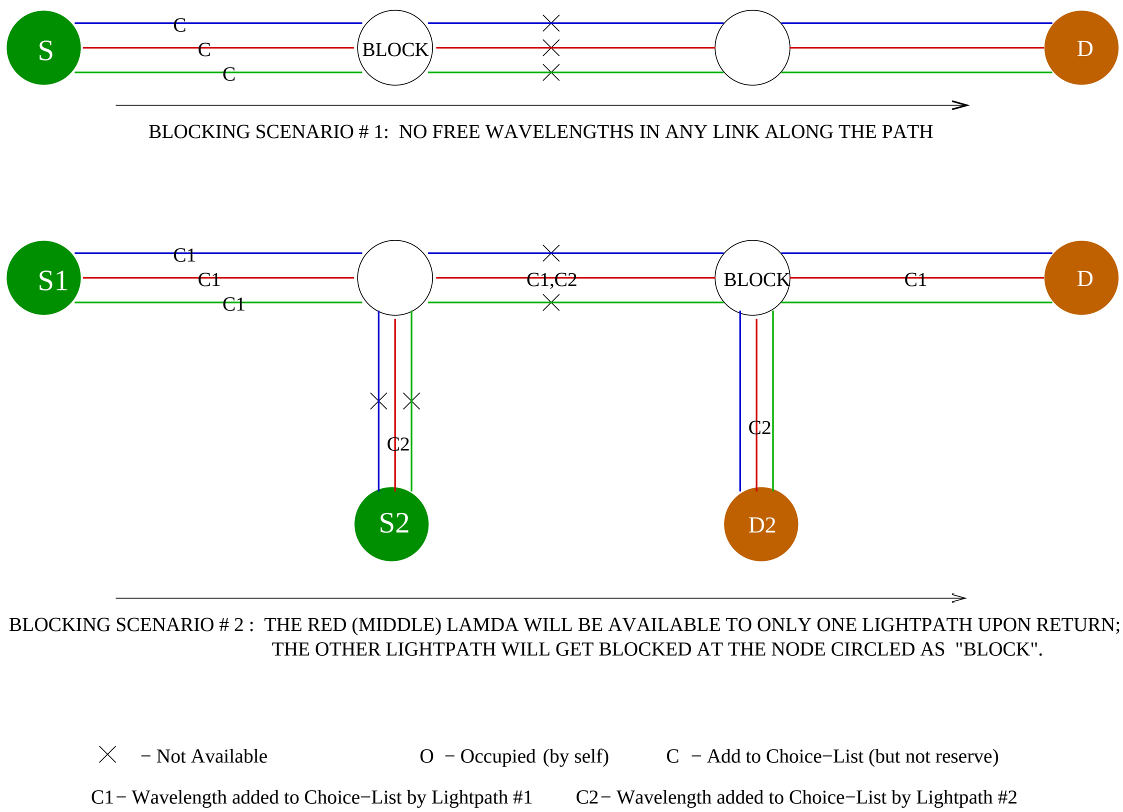

Figure 2.12: Blocking in BRP - Subset Version

- Inability to satisfy the wavelength continuity constraint due to the current state of reservation or usage of wavelengths by other lightpaths along the links of our interest. The causes are similar to as in FRP (see Figure 2.11).
- The selected wavelength by the destination node is no longer available in one of the links along the path, for the selected wavelength has been used in setting up another lightpath request by the time the signaling message of interest makes its way to the link. This scenario is more likely in non-zero delay scenarios.

The first and the last points above are illustrated by the diagram in Figure 2.12.

5. **Wavelength Assignment Policy:** The wavelength assignment policies that we describe below are basically heuristics used to decide which wavelength to select among many available wavelengths for a given lightpath request. We describe some of the well-known heuristics.

- *First-fit:* The wavelengths are ordered in a pre-determined way and the first wavelength that is available, in that order, is chosen. This heuristic requires the node to be only locally aware as opposed to some schemes (seen below in this section) where global network knowledge is required to choose the wavelength. The first-fit heuristic is simple and performs reasonably well relative to other heuristics described below.

- *Random:* Out of the many wavelengths available, a wavelength is chosen randomly. Again, global network knowledge is not required of the node.

- *Most-used:* In this heuristic, global network knowledge is required of the node. The node responsible for selection of the wavelength is required to find out what wavelength is used most throughout the network. It then orders the wavelength to be chosen in the order of most-used and selects an available wavelength according to that order. The idea is to pack the wavelength usage and increase the utilization of certain wavelengths. Most-used heuristics perform well but they are more complicated to implement as they require global network knowledge of the wavelength utilization.

Chapter 3

Establishing Lightpaths - Routing & Wavelength Assignment (RWA) Problem

3.1 Introduction

Establishing lightpaths involves selecting a route and choosing a wavelength along that route. If the nodes along the route selected are incapable of converting wavelengths, then it must be assured that a wavelength chosen for the lightpath is available on all links along the chosen route. This is called the wavelength continuity constraint. The problem of selecting an optimal route and a wavelength for a lightpath such that the network throughput is maximized or minimize blocking probability is a tightly coupled problem and has no polynomial-time analytical solutions [6]. This problem is called the RWA (Routing and Wavelength Assignment) problem to which the reader was introduced briefly before. Since the tightly coupled RWA problem cannot be analytically solved, it is a general practice to de-couple the problem and try to solve the routing and wavelength assignment problems separately. While the routing sub-problem may be optimized, the wavelength assignment sub-problem turns out to be NP-hard for non-trivial topologies. Thus heuristics-based approach is adopted for wavelength assignment sub-problem and near-optimal solutions for it are sought. The near-optimized solutions for the routing and wavelength assignment sub-problems respectively may then be combined. Ofcourse, this approach need not necessarily lead to the globally optimized solution of the original coupled RWA problem. As a result, there are various heuristics suggested for the wavelength assignment problem and various routing schemes used for the routing problem. The performance of such assignments coupled with routing schemes is then studied by simulation.

3.2 Literature Review on RWA Problem

To begin with, the RWA problem can be viewed as a part of the virtual topology design problem, which has four parts [16]. The first part wishes to determine the virtual topology that needs to be imposed on the physical topology by specifying the lightpaths that need to be established between source and destination nodes. Due to resource restrictions, it might not be possible to establish a lightpath between each source-destination pair. The second and third part of the virtual topology design problem are the Routing problem and Wavelength Assignment problem respectively. When considered together, the Routing and Wavelength Assignment problem seeks to address the question: Given a set of lightpaths that need to be established, what is the best way to achieve it? The "best way" is mathematically formulated as an objective function; it could be the minimum number of wavelengths needed to establish the given set of lightpaths or it could be to minimize the blocking probability given a bound on the number of wavelengths (this bound is obviously less than the minimum number of wavelengths we would have required had we assumed an unbounded number of wavelengths were available). The last part of the virtual topology design problem is to effectively 'groom' the traffic (i.e to send the packets from the source to the destination using the virtual topology obtained so far), thereby hoping to maximize the network utilization. The RWA problem (parts 2 and 3 of virtual topolody design problem) is a very tightly-coupled problem.

The RWA problem is known to be NP-hard with variants of it, like the Static Lightpath Establishment (SLE) problem where a bounded number of wavelengths need to be assigned to the lightpaths, given the routes and lightpaths to be established, being NPcomplete [6]. Its well understood in the academic community that there exists no optimal polynomal time algorithm, as of today, for the RWA problem. Hence there is a need for heuristics for both routing and wavelength assignment sub-problems.

Given the background of RWA problem as explained above, we would like to review some of the studies done in the literature keeping the traffic demands as dynamic (i.e traffic demands not known apriori).

Since the RWA problem is tightly coupled, sets of different combinations of routing schemes and a wavelength assignment policies need to be determined for good performance. Performance may be characterized by using different measures of interest. They include blocking probability, average set-up time, utilization of links and so on. Blocking probability is often used as the performance measure as it is directly related to the revenue of the service providers. While blocking probability may always be computed by simulation, analytical expressions are desired. Analytical models for computing blocking probability in all-optical networks ([2],[4]), multi-fiber WDM networks([11]), optical networks with non-Poisson traffic ([20]) have been studied in the past. Path-decomposition techniques exist ([15],[25]) to estimate the blocking probability in wavelength-routed networks.

The typical approach, as mentioned earlier, to solve the RWA problem is to decouple it and try to obtain near-optimal solutions for the routing and wavelength assignment sub-problems and then to combine their solutions. The simplest choice is to go ahead with Shortest Path (SP) routing scheme coupled with first-fit wavelength assignment policy which is distributed in nature. One can also use an Alternate Shortest Path (ASP) routing scheme with more than one pre-computed routes for each source-destination pair. The performance of SP and ASP routing schemes have been studied in [4],[10],[8] and [14]. Clearly, static routing schemes lead to loading the same paths with traffic over and over as time proceeds. Thus, despite having alternate longer paths that maybe less loaded, the traffic is still routed through these pre-computed shortest paths where the metric typically used is hop count. Hence the idea of adaptive RWA came about. Here, the network state information is either exchanged periodically between routers or obtained on a per-need basis by sending probing messages. Based on this “latest” information, a route and a suitable wavelength is searched for to make a connection for the lightpath request. Adaptive routing schemes fall under the general category of dynamic routing. Different flavors of dynamic routing, including adaptive routing, have been studied in [12],[13] and [7]. A review and comparison of various wavelength-assignment policies has been done in [23] and [24] respectively.

While the above review refers to routing and wavelength assignment schemes, wavelength reservation schemes are an important aspect of this problem, especially when the delay experienced by these signaling messages is no longer negligible, something which we are interested in investigating here. A good study on various wavelength reservation schemes has been done in [18]. We have borrowed the ideas for FRP, BRP, FRP-SS and BRP-SS reservation schemes described earlier from [18] (although they have different names in that work) for our study here. The work in [18] does study the effect of signaling delays but only for the case of random wavelength assignment policy with fixed-routing scheme. The topologies are randomly generated and do not include any regular topologies. The focus there lies more on the effect of various wavelength reservation schemes on performance than on the effect that a combination of routing, wavelength assignment and wavelength reservation schemes have as we have done here in this thesis.

Past studies ([7]) contains work that takes into account the propagation delay involved in the signaling messages. However their study had randomly generated topologies and did not include any regular or NSFNET topologies. Their focus lie more on finding a way to characterise the cost of a link than comparing wavelength assignment policies and reservation schemes. They sought a combination of static and dynamic metrics, such as hop count, propagation delay, available wavelengths and total wavelengths to come up with a cost-function for estimating the costs of a link in a dynamic scenario. [22] studied the forward and backward wavelength reservation schemes with propagation delay involved. They used a propagation delay of 1 unit per link, an average holding time (connection duration) ranging from 50 to 500 time units and studied mesh torus networks that ranged from a size 4×4 to 16×16 . Their paper did not explicitly indicate what wavelength assignment policy they used. Their wording, however, seemed to suggest that the random assignment policy was being used. Their performance measure of interest was throughput (average rate of bits per second).

3.3 Choice of Signaling protocol used in this study

Since the subject of this thesis is to evaluate the performance of establishing lightpaths in an environment where signaling messages experience a non-negligible delay, the first problem was one of choosing a signaling protocol whose semantics we wished to simulate. There are two signaling protocols, standardized by IETF, whose semantics maybe used to establish lightpaths in our simulation study. The protocols are Resource Reservation Protocol -Traffic Engineering (RSVP-TE) and Constraint-Routed Label Distribution Protocol (CR-LDP). While the former has been extended to encompass signaling functions, the latter was designed to be a signaling protocol from start. While the industry is divided on which protocol will eventually win, some vendors are developing products supporting both protocols. We choose the CR-LDP semantics here as it is forms an integral part of the MPLS solution in IP networks.

3.3.1 Multi-Protocol Label Switching (MPLS)

The goal of certain telecom equipment manufacturers was to speed up the look-up times in the router tables by tagging the incoming and outgoing packets. Meanwhile, faster algorithms were devised to reduce the router table look-up times. However, the effort of providing an architecture of tags, and how they may be used did not go a waste. These ideas were instead used to support service guarantees in otherwise unreliable and best-effort IP networks. The MPLS working group was thus established under the aegis of the IETF. MPLS [17] is essentially an architecture that describes the functionalities that networks need to support so that they may be used for Traffic Engineering that in turn may be used to provide Quality of Service (QoS). MPLS become very useful for it separates the function of routing (by assigning the packet to a forward equivalence class, FEC) and forwarding (by Label-switching). Since the fundamental entities used to achieve the above are tags, or labels as they are called in MPLS domain, lets take a brief look at them first.

3.3.2 Label Distribution Protocol (LDP)

A label is a short, fixed length locally significant identifier, which is used to identify a FEC. An IP packet is assigned to a particular FEC (typically based on the packets' network layer destination address). Service differentiation may be introduced in the MPLS networks using this notion of FEC. To decide the value of the labels to be given to the IP packets, the MPLS working group came up with a Label Distribution Protocol (LDP) [1] such that the labels are distributed to every node in a network. Since the labels are significant only locally, the labels corresponding to the same FEC may be different in different networks and even nodes in the same networks. The LDP specifies the behavior of a router depending on whether it is upstream or downstream [1]. The routers that forward packets based on labels are called the label-switched routers (LSR). When a packet is forwarded along a set of routers using these labels, the path formed by the set of routers is called a label-switched path (LSP).

3.3.3 Constrained-Routing Label Distribution Protocol (CR-LDP)

The Constrained-based Label Distribution Protocol (CR-LDP) [9] is an end-to-end signaling protocol designed for IP/MPLS networks. CR-LDP extends the original MPLS Label Distribution Protocol (LDP) by considering constraints beyond normal routing information and thus providing Quality of Service (QoS) and Traffic Engineering (TE) capabilities. CR-LDP provides the ability to establish explicitly routed paths by allowing an ingress LSR to pre-calculate the list of nodes that specify the route to the egress LSR. Since it uses similar mechanisms as LDP, CR-LDP can be used for both regular IP routed LSPs and constrained-based routed CR-LSPs. CR-LDP forms an inherent part of the MPLS protocol suite as opposed to RSVP-TE, which has its roots in the Integrated Service (Intserv) QoS model. The signaling requests reservation at the micro-flow level between two end stations and is not per connection. RSVP, which is strictly a control plane protocol and used to reserve resources for sessions rather than sources, had to be extended to RSVP-TE to support MPLS networks. Hence we use CR-LDP as it was natively designed for MPLS networks.

CR-LSP Establishment: The most important function of CR-LDP we are concerned with is the path establishment procedure which is illustrated in Figure 3.1 and described below.

- The ingress LSR-A computes an explicit route to LSR-D and sends its setup (called LABEL REQUEST) message to its next computed hop, namely, LSR-B. LSR-A might need to check for availability of resources in its outgoing link before sending it to LSR-B.

- LSR-B would forward the setup message to LSR-C after performing checks as described above. Every intermediate node would check if it can support the traffic parameters included in the setup message. This continues till the setup message reaches destination, here, LSR-D.

- LSR-D, upon reception of the setup message will check if it can support the resources asked for. If it can, it will send its response (called LABEL MAPPING) message containing the chosen label (wavelength in our case) to the first node in the reverse direction, here, LSR-C.
- The response continues to be forwarded in the reverse direction by the intermediate LSRs' with each capable of providing its own label (if needs be), till it reaches the source when the lightpath is deemed established.

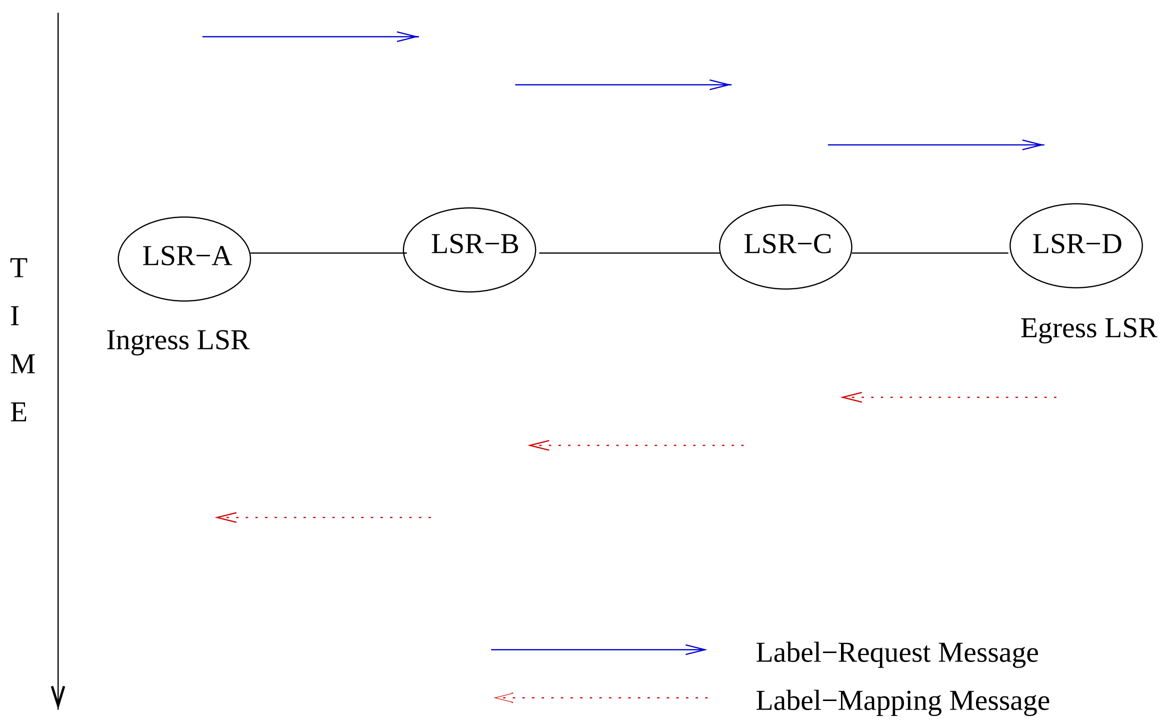

Figure 3.1: Constrained-Routed Label Switched Path - Establishment (Message exchange)

CR-LSP teardown requires explicit messaging from the ingress to the egress LSR. The downstream LSR releases the resources upon receipt of the teardown message. The teardown procedure is not shown here.

Chapter 4

Simulator Overview

This chapter contains high-level overview of the simulator that was developed for the study involved in this thesis work. The simulator was developed for optical WANs and it is capable of handling the following:

- Arbitrary input physical topologies that involve multi-fiber links.
- Static, Alternate (with maximum of 10 re-attempts) and Dynamic routing schemes.
- FRP-SS and BRP-SS wavelength reservation protocol.
- First-fit, Random, and Most-Used policies.

4.1 List of Modules and Class Hierarchy

The simulator may broadly be broken into in four 'functional' modules.

1. Topology Constructor Module
2. Routing Module.
3. Wavelength Reservation and Assignment module.
4. Event-handling (Simulator) Module.

Each module shall be described, in subsequent sections, using the following template:

- Functional description: This will include the functionality supported by, and the responsibilities of the module.
- List of important functions/methods: This will include a brief description of the important input parameters to the function/method and their important output values, in addition to describing the primary responsibility of the method. These methods may be present in different classes. The idea behind this exercise of listing the methods is merely to group the methods by core functionality.

The class hierarchy with some comments, if deemed necessary, will be provided next.

1. Base Classes:
    - Simu-object: This class is the parent class of the important simulation objects, such as Node, Link and Lightpath, to be used in the simulator. This is the only class in the simulator which is inherited from.
    - Routing-Module: This class is responsible for dynamic routing, forwarding and pre-computing static routes.
    - Path: This class stores the path to be traversed by a Lightpath object.
    - Wavelength-Assignment: This class is responsible for the reservation, selection and confirmation of wavelengths.
    - Simulator: This class is the main part of the Event-handling module and is responsible for simulating the CR-LDP semantics thereby ensuring how the signaling messages behave.
    - Heap: This class, implements the heap structure for the event-list. The event-list stores pointers to objects belonging to Simu-object class (and hence its inherited ones too) as its event-objects.
    - Pair-stats: This class collects the statistics generated by individual source-destination pairs.
    - Stats-collector: This class, collects, manipulates and stores the statistics generated by Pair-stats class.

- Wavelength: This class represents the wavelength objects (in each link).
- Random: This class generates the random number according to a specified distribution like exponential, uniform etc. The class is built such that it may be easily modified to accomodate other distribution.
- Topology-Constructor: This class generates the physical topology representation after parsing an input file.

2. Inherited Classes:

   - Node: This class in inherited from Simu-object class. The nodes in the network are represented as Node objects.
   - Link: This class in inherited from Simu-object class. The links in the network are represented as Link objects.
   - Lightpath: This class in inherited from Simu-object class. The signaling messages used to establish lightpaths in the network are represented as Lightpath objects, which are instances of Lightpath class. The terms call and lightpath may be used inter-exchangeably in our discussion.

As pointed out above, the signaling message is represented by a Lightpath object. Important information like the path to be followed, the current node being visited etc, is stored, updated, and accessed using this object, as the signaling message traverses from one node to another along the path.

4.2 Topology Constructor Module

- Functional description: The Topology Constructor Module is primarily entrusted with the responsibility of parsing an input file, reading the system parameters and information related to nodes and links, thereby storing all the input information needed for the simulation. Using this information, it then creates the Node objects and the Link objects and stores them in a global array. The Node class and the Link class are provided with pointers so that an association between the links and the nodes are established. Thus the Topology Constructor Module generates the representation of a physical topology in the form of a global array.

- List of important methods in this module:

  1. Create-nodes() : The method takes in a string containing the input information about the nodes and the node number that need to be assigned to the node. The method parses the string and creates the Node object with the parameters contained (for the node) in the string. The Topology Constructor module creates all nodes in the graph before creating any other object.

  2. Create-links() : The method takes in a input string containing the node numbers of the two nodes that form the ends of the directed link to be created and the delay information associated with the link. The total number of wavelengths that a link must support is given as an input. After parsing the information from the string, it creates the Link object with the necessary parameters passed and makes the pointers (defined in the Link class) of the Link object point to the 'head' and 'tail' Node (the two ends of the link) objects. The diagram in Figure 4.1 identifies the 'head' and the 'tail' node of a link. The terms 'head' and 'tail' are used in an opposite sense to the conventional understanding of 'head' and 'tail'. The 'head node' may be loosely interpreted as the 'source node' and the 'tail node' as the 'destination node' when we look at the data flow via the directed link.

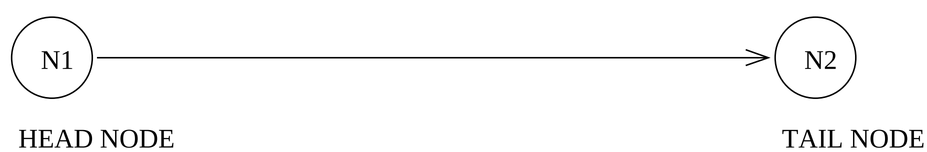

Figure 4.1: Example of a directed link

  3. Create-link-from-node() : This method exists in Node class. It takes in a string and pointer to a Link object as the main inputs. The string identifies whether the node in question is a head or a tail to the input link. There are two lists (defined in Node class) provided to each Node object; one is a list of pointer to links for which the node acts as a 'head node' and the other is a list of pointer to links for which the node acts as a 'tail node'. Thus, this method adds the input pointer to the Link object to either the 'head' list or 'tail' list depending on the value of the string. Thus this method helps a Node object to classify the links it

is attached to, as incoming links and outgoing links.

Thus with these three methods, the important information regarding the physical topology information is captured. There are other variables and methods in the module that store and access the values representing the kind of traffic to be simulated, routing scheme to be employed etc.

4.3 Routing Module

- Functional description: The Routing Module is responsible for discovering a route given a source-destination pair (this route information is embedded as a Path object in the signaling message represented as a Lightpath object), forwarding the signaling message to the next node in path, storing link-disjoint paths for all source-destination pairs in a static routing scheme and for reversing a route so that the signaling message returns to the source node. This module also calculates the diameter of the physical topology in two ways; one using hop count as the metric, and the other by setting it equal to the maximum one-way delay or round-trip-time (RTT) that signaling messages undergo while traversing the network. Thus, in the latter case, the diameter of the network has units of time.

- List of important methods in this module:

  1. Dijkstra-algo() : As the name suggests, this method implements the Dijkstra's algorithm to determine the shortest path from a source node to a destination node. The metric employed is 'link cost', which is an attribute of a Link object and may be updated as simulation proceeds. This method takes in a pointer to source and destination Node objects and returns a pointer to a Path object that contains the shortest path. The path is represented in the form of a linked-list containing Node and Link objects that need to be visited in the forward direction.

  2. Static-route() : This method computes and stores the shortest path between all pairs of source destination pairs before the simulation begins. It takes the number of reattempts as the input and computes that many link-disjoint paths for every source destination pair. This method is executed in both static and

dynamic routing schemes, as it is also useful in estimating the diameter of the graph, either in terms of hop count or round-trip-times (RTT).

3. Route(): This method takes in the signaling message, represented by a Lightpath object, as the input and embeds the path (either statically computed or dynamically computed) for the source destination pair that the Lightpath object contains.
4. Forward() : This method takes in the signaling message, represented by a Lightpath object, as the input and increments the appropriate pointers within the Lightpath object.The Lightpath object, thus, gets handled at the next Node and Link objects contained in the Path object.
5. Route-reverse() : This method takes in the signaling message, represented by a Lightpath object, as the input and reverses the path (source to destination OR destination to source, depending on the situation) so that the Lightpath object could now be forwarded in the opposite direction. This method is typically called either at the destination node or at a node where the lightpath request fails. The route is thus reversed allowing the Lightpath object to inform the previous nodes in the path of the failure.
6. Calculate-and-set-diameter() : This method uses the output of the Static-Route() method and computes the diameter of the graph (physical topology) in terms of hop count and round-trip-time (RTT).

4.4 Wavelength Reservation and Assignment module

- Functional Description: The Wavelength Reservation and Assignment module serves as the module that does the wavelength reservation, wavelength selection (thereby helping the destination node make a decision), and wavelength confirmation at a particular link along the path. It is also responsible for releasing the reserved/confirmed wavelengths (for example, when the teardown message is sent), thus making them available, in the links that are used by different lightpaths.
- List of important methods in this module:

1. Reserve() : This method takes in the signaling message, represented by a Lightpath object, as the input. It reserves the wavelength in the current link or probes for availability of wavelengths in the current link depending on whether the wavelength reservation protocol is FRP or BRP respectively. It then adds or prunes the list of reserved or available wavelengths as the signaling message proceeds. The list of reserved or available wavelengths is stored and updated in the Lightpath object as the signaling message is simulated to proceed to the destination.
2. Select-lamda() : The destination node refers to this method upon reception of the signaling message (Lightpath object). This method then selects one wavelength out of the many options presented in the form of a list of reserved (in case of FRP and its variants) or available (in case of BRP and its variants) wavelengths and updates the signaling messages.
3. Confirm() : This method seeks to confirm the usage of the selected wavelength for the current lightpath request in question. Thus every wavelength belonging to a link, upon confirmation, is given an ownership identification number that belongs to the lightpath request. The wavelength in that link is then said to be *owned* by that lightpath. A wavelength in a link may be owned by only one lightpath at any given time. Thus the further manipulation of the status of this wavelength is possible only by that Lightpath object. However, a lightpath request may own many wavelengths in a link at the same time (e.g. more than one wavelength in a link reserved by lightpath request in FRP schemes).
4. Release-all-lamdas() : This method, as the name suggests, releases the wavelengths in a link owned by the lightpath of interest, thus making them available to other lightpaths. This method is typically called when a lightpath is torn down or when a lightpath has reserved many wavelenghts in a link for itself and needs to release all but one (the one that is selected) wavelengths in that link.

4.5 Event-Handling Module

- Functional description: The Event-handling module handles the events such as

1. Generation of a lightpath request.
2. Arrival of a lightpath request at a certain node.
3. Teardown event of a lightpath already established.
4. Denial of a lightpath request.
5. Routing and forwarding the lightpath request to the next node in path.

This module interfaces with the Topology Constructor Module, the Routing module and the Wavelength Reservation and Assignment module while handling an event. The event-objects, represented by Lightpath objects, are inserted into and removed from an event-list implemented in the form of a Heap object.

- List of important methods in this module:

  1. Set-params-new-call() : This method sets the parameters, using random number generation, for a new call (lightpath request). The parameters include the source node, destination node and the start time. Call identification number (used to uniquely identify a lightpath request) is increased sequentially. The choice of source and destination node numbers may be manipulated by choosing an appropriate distribution (e.g. Uniform distribution). This call is represented as a Lightpath object.
  2. Set-params-teardown() : This method draws a teardown time randomly and is used to set the teardown time of the current lightpath object in question. Again, an appropriate distribution, like the exponential, normal or uniform distribution may be used to determine this time.
  3. Check-if-dest-node() : This method returns true if the signaling message, represented by a Lightpath object, has reached the destination node.
  4. Check-if-source-node() : This method returns true if the signaling message, represented by a Lightpath object, has reached the source node (upon return).
  5. Process-calls() : This method is the heart of the Event-Handling module and indeed of the entire Simulator! It implements the 2-phased signaling protocol (CR-LDP semantics) for this study. The simulator extracts the Lightpath object (event-object) from the event-list heap and then upon completion of handling

of the Lightpath object, inserts it back into the heap or destroys it, if deemed appropriate. This method handles six specific events:

(a) *New-call-request:* This is the first time a Lightpath object, once created, is handled by the Event-Handling Module. A route for the lightpath request, if it exists (if the route does not exist, the Lightpath object is destroyed), is found by calling appropriate methods in the Routing Module. The source node and its outgoing link are then probed to find whether at least one free wavelength exists to support the lightpath. If it exists, the lightpath is forwarded to the next link using appropriate methods in the Routing Module. If not a single free wavelength is available, the Lightpath object is marked as 'Request-denied' and is inserted into the heap. It is then handled as indicated in the item 'Request-denied' described below. The list of wavelengths to be used gets updated every time the Lightpath object visits a node in its path. This list is determined by using appropriate methods in the Wavelength Reservation and Assignment Module. After this, the Lightpath object is now labeled as a 'Transit-call-request'. Upon completion of handling of this Lightpath object, a new Lightpath object representing future lightpath request (i.e future traffic) is then generated. Both the current and future Lightpath objects are then inserted into the heap. The next processing time for the current Lightpath object is set to the current time plus the delay it is expected to experience while traversing the current outgoing link. The new Lightpath object will have its processing time equal its start time, which in turn is equal to the current time plus the inter-arrival time computed according to the traffic distribution used.

(b) *Transit-call-request:* This event indicates that the lightpath is not being handled for the first time in the forward direction. The availability of wavelengths is checked for and if successful, it is forwarded to the next node and the status remains unchanged. This continues till the lightpath reaches the destination node. Upon reaching the destination node, the status of the Lightpath object is changed to 'New-accepted-ack'. If, at any point in the path, no wavelength (as per the list of wavelengths to be used) is available, the status of the Lightpath object is changed to 'Request-denied'. Once the status is set at a particular node, the Lightpath object is then inserted in the event-list heap. Its next processing time is set to the current time plus the delay it is expected to experience while traversing the current outgoing link.

(c) *Request-denied:* This event indicates that the lightpath request has been rejected for lack of resources (e.g. no path exists, no free wavelength available etc). If the lightpath was rejected in the forward (going towards destination node) direction, the lightpath is then made to travel in the reverse direction by a route reversal, towards the source node, freeing up the resources along the way held up by that lightpath. If the lightpath was rejected in the backward (returning to the source node) direction, the message continues towards the source direction and a copy of the Lightpath object is sent along the path towards the destination node by implementing route reversal. The route reversals are achieved by using the appropriate methods in Routing Module. If the rejected Lightpath object reaches the source node, the statistics are updated and the Lightpath object destroyed in a dynamic routing scheme. However, in case of static routing scheme with a finite number of reattempts, instead of destroying the Lightpath object, the lightpath request is given a new lease of life. A new path that is link-disjoint with respect to the earlier paths is provided. The status is set to 'New-call-request' and the Lightpath object inserted into the heap. The Lightpath object has its next processing time set to current processing time plus the delay it will experience in its current outgoing link.

(d) *New-accepted-ack:* This event indicates that the lightpath has reached the destination node with at least one wavelength available for the entire path. The destination node chooses the wavelength out of the many possible choices according to a wavelength assignment policy. This is implemented by making appropriate calls to the methods in the Wavelength Reservation and Assignment Module. Having selected the wavelength, the information is stored in the Lightpath object in the form of a variable representing the selected wavelength. The route is reversed and the Lightpath object is sent towards the source node. The status of the lightpath object is changed to

'Transit-accepted-ack' and forwarded to the first node in the reversed path for confirmation. It is then inserted in the event-list heap. As always, the Lightpath object has its next processing time set to current processing time plus the delay it will experience in its current outgoing link.

(e) *Transit-accepted-ack:* This event implies that the Lightpath object has a wavelength selected for it and is en route to the source node. Attempts are made at every hop to confirm the selected wavelength and release other wavelengths that may have been reserved for this lightpath request. Failure to do so at any stage along the path results in the lightpath request being rejected and the status is set to 'Request-denied'. If the lightpath request successfully reaches the source node, the lightpath is deemed established on the selected wavelength and a teardown time for the lightpath drawn randomly. The status of the lightpath is then set to 'Teardown' and the Lightpath object inserted in the event-list heap. The Lightpath object has its next processing time set to current processing time plus the teardown time (i.e. the holding time) drawn randomly as per an appropriate distribution.

(f) *Teardown:* The lightpath object is made to traverse from source to destination releasing the wavelength it occupied along the way. Upon reaching the destination, the success of the lightpath establishment is accounted for and the Lightpath object destroyed.

Chapter 5

Results & Discussion

Simulations were run for different routing schemes, wavelength assignment policies, traffic loads for 5×5-torus topology shown in Figure 5.1, 12-node ring topology shown in Figure 5.2 and NSFNET topology shown in Figure 5.3. The signaling delay, for 5×5-torus and ring topology, in each link was assumed to be 4 units. The diameter of such a network was thus equal to 6 and 11 respectively in terms of hop count.Clearly, the diameter of the network also equaled 24 and 44 time units respectively. It was a measure of the maximum signaling delay that could be faced by a signaling message in the network.

Each curve presented in the graphs that follow in this chapter is a result of 30 different runs, each with a different random seed. The values of the average holding time ($1/\mu$) used were 0.1, 1, and 10 times the one-way diameter (calculated in terms of delay) of the network. Traffic load (λ/μ) values used were 0.01, 0.25, and 1.0 per source-destination pair. The first-fit, random and most-used heuristics were the wavelength assignment policies simulated in conjunction with the BRP-SS wavelength reservation scheme. Only the first-fit assignment policy was simulated for the FRP-SS wavelength reservation scheme. The graphs, in the sections to follow, have names and/or values of the topology, the wavelength reservation scheme, the wavelength assignment policy, the routing scheme, the traffic load (for the entire network as opposed to per (s,d) pair), and the holding times indicated on its top. The number of wavelengths (W) used in all simulations was 10. The captions of the figures provide additional comment if deemed necessary. Before we discuss the results, a brief summary of the findings is presented below.

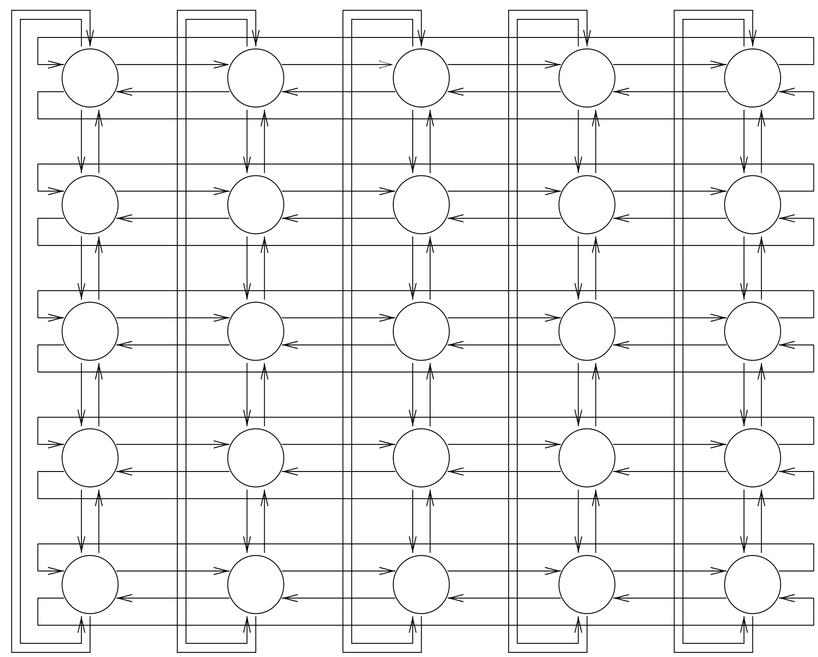

Figure 5.1: 5×5-Torus Topology

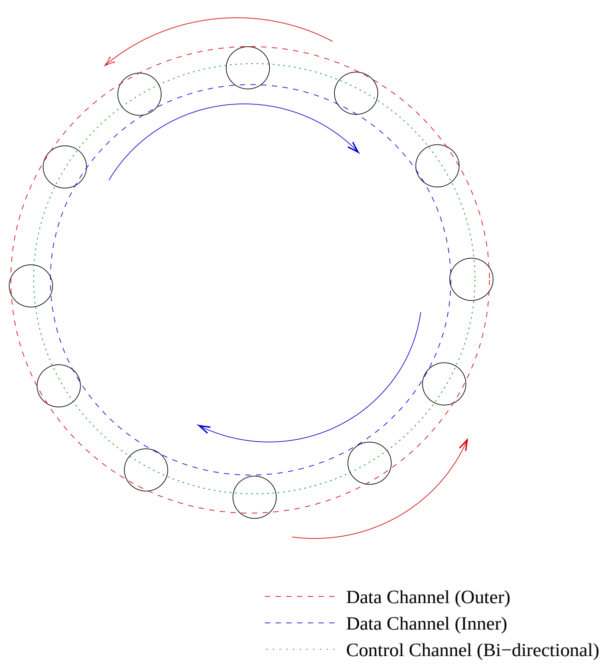

Figure 5.2: 12-node Ring Topology

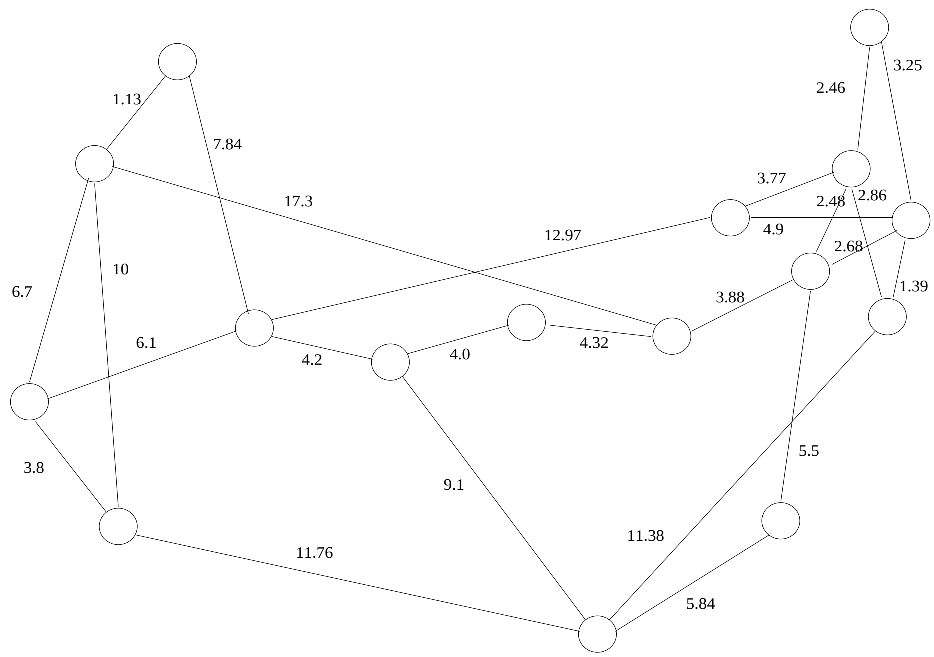

Figure 5.3: NSFNET Topology; Approximate propagation delays are given based on distance between cities

- As the average holding time ($1/\mu$) was increased from 0.1 to 10 times the one-way diameter of the network, the blocking probability values fell for all hop lengths for a given routing scheme, wavelength reservation scheme, wavelength assignment policy and a traffic load. This trend was observed across varying traffic loads.
- The random wavelength assignment policy out-performed the first-fit policy for the topologies studied under conditions of varying traffic loads, wavelength reservation schemes and holding time of lightpaths relative to the signaling delays. This result is contrary to the ones established so far but was not entirely unexpected.
- Alternate routing scheme does not necessarily perform better than fixed routing scheme in all cases; especially when the average holding time is less than the order of magintude of the diameter (computed in terms of signaling delay) of the network. This was observed across the topologies we studied and was an unexpected result.
- BRP-SS scheme performs better than FRP-SS when first-fit wavelength assignment policy is used but the difference becomes less as the traffic load decreases or as the average holding time increases relative to the signaling delays.

- For a given wavelength reservation scheme, wavelength reservation policy and holding time, as the traffic load increases, the blocking probability values increase for all hop lengths as expected.

We now look at each one of the above results in detail.

5.1 Effect of average holding time ($1/\mu$) relative to signaling delays

The holding time for a lightpath (with the average being $1/\mu$) is defined as the time from which the lightpath gets established to the time when the teardown message is issued. The holding time is expressed as a factor of the one-way diameter (expressed in delay terms and hence time units) of the network. It was found that as the holding time for a lightpath increases relative to the signaling delays, the blocking probability decreases for a given routing scheme, a given wavelength reservation scheme, wavelength assignment policy and traffic load. This can be justified by realising that as the holding time tends to get shorter relative to the signaling delays, the teardown request is issued quicker thereby leading to a dynamic situation in the network. Thus, with lower holding times relative to the signaling delays, the state of the network changes increasingly faster and as a result more blocking occurs. We show results only for NSFNET and 5×5-torus networks; similar tendencies were found in ring topologies. Figure 5.4 and Figure 5.5 show the results for fixed routing while Figure 5.6 shows similar trends for alternate routing in NSFNET topologies. Figure 5.7 and Figure 5.8 show the results for 5×5-torus topologies. The graphs indicate that the trend observed is valid across wavelength assignment policies, and reservation schemes.

5.2 Random Versus First-Fit Assignment Policies

The first fit policy is known to perform better than the random wavelength assignment policy ([6], [10]) when signaling delays are negligible. However with a non-negligible amount of signaling delay, our results show the contrary; random policy performs better than first-fit policy for short hop lengths. Figure 5.9 and Figure 5.10 present results for

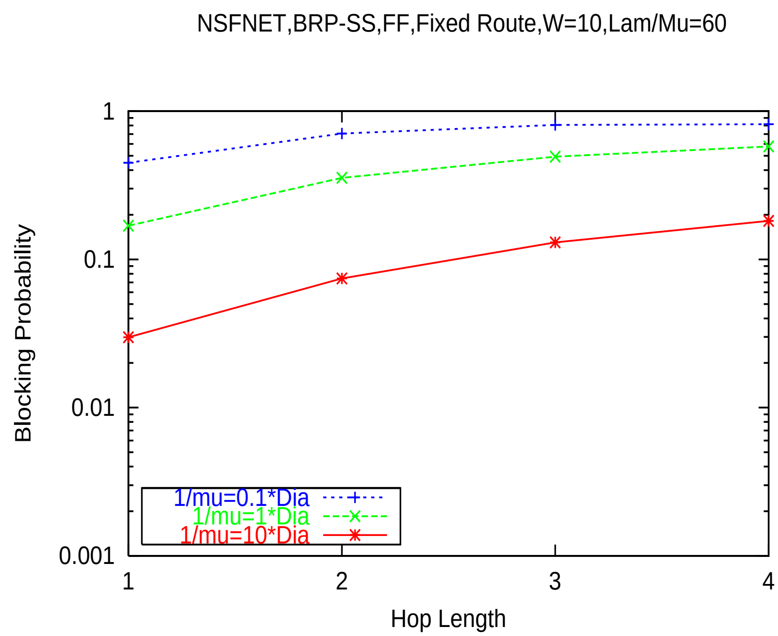

Figure 5.4: Blocking Probability increases as average holding time $(1/\mu)$ decreases relative to the signaling delays; FF = First Fit, Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

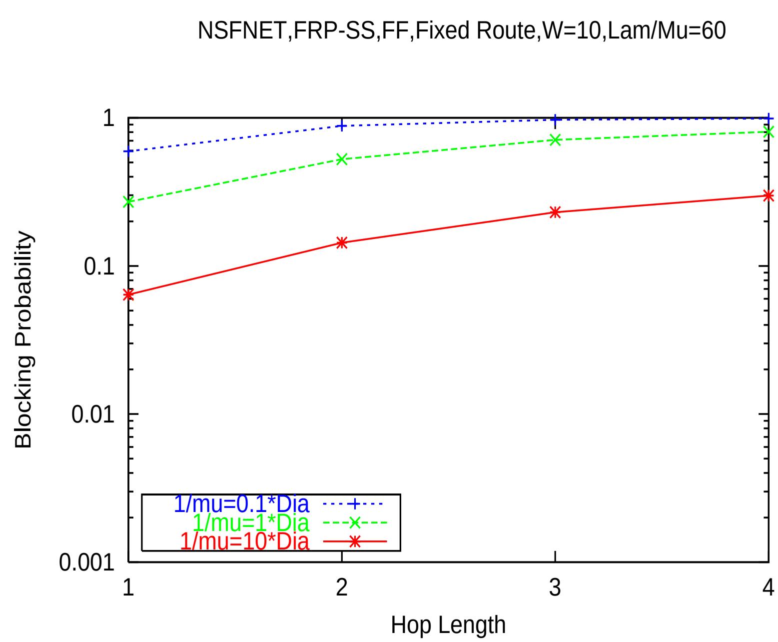

Figure 5.5: Blocking Probability increases as average holding time $(1/\mu)$ decreases relative to the signaling delays; FF = First Fit, Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

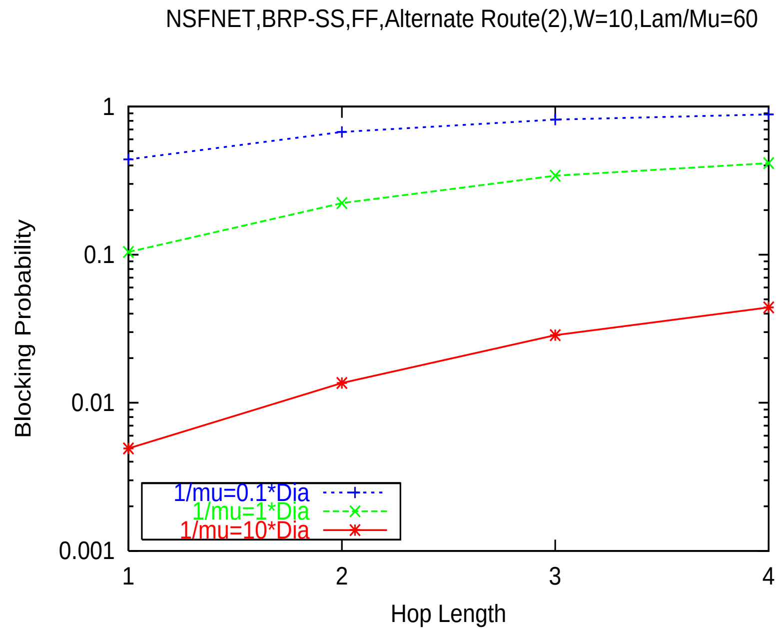

Figure 5.6: Blocking Probability increases as average holding time $(1/\mu)$ decreases relative to the signaling delays; FF = First Fit, Traffic load (λ/μ) = 0.25 per (S,D) pair, Alternate Route with 2 pre-computed paths

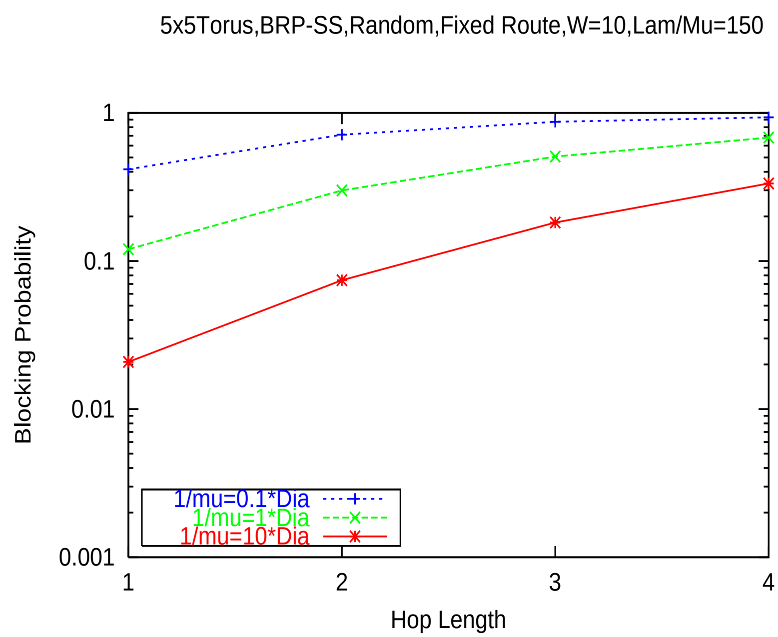

Figure 5.7: Blocking Probability increases as average holding time $(1/\mu)$ decreases relative to the signaling delays; Traffic load (λ/μ) = 0.25 per (S,D) pair

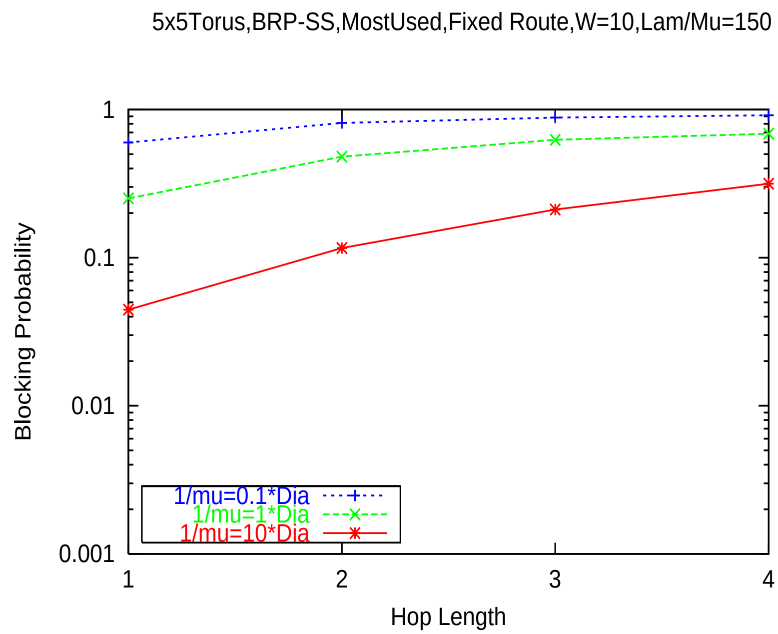

Figure 5.8: Blocking Probability increases as average holding time ($1/\mu$) decreases relative to the signaling delays; Traffic load (λ/μ) = 0.25 per (S,D) pair

5×5-torus networks and Figure 5.11 through Figure 5.14 for NSFNET topologies. Similar trends are observed for ring topologies but are not shown here. It may be argued that since the first fit policy attempts to utilize the available wavelengths based on a certain pre-defined order, the destination nodes of two different lightpath requests traversing a common link/links tend to select the same (if available) wavelength to set up their respective lightpaths. Thus upon return of the signaling messages to the source node (as in the case of BRP-SS scheme), only one of the lightpath request would get established using that wavelength and the other lightpath gets blocked because the wavelength of interest has been occupied by the established lightpath. Random policy leads the two destination nodes, in the above example, to spread that risk of selecting a common wavelength due to precisely the 'random' nature of the policy! Thus, we notice in Figure 5.9 through Figure 5.14 that random policies perform consistently better than their first-fit counterparts for short hop lengths, across different traffic loads, different holding times, and routing schemes. As hop lengths increase, the effectiveness of the random policy decreases. This is because,in random policy, the wavelengths used up are spread out leading to difficulties in satisfying the wavelength continuity constraint for longer hop length lightpaths. Despite this, the random

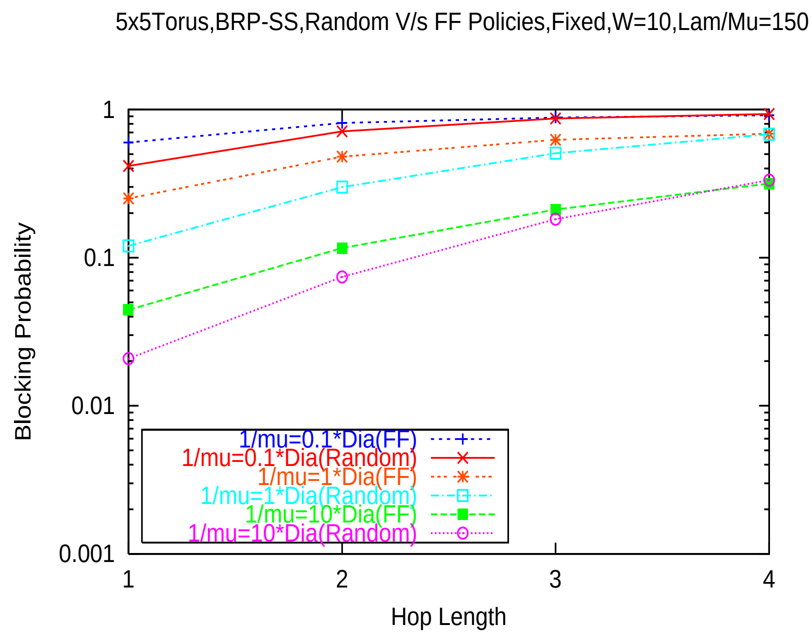

Figure 5.9: Compare Random Versus First-Fit Assignment Policy; BRP Scheme; 5×5-torus; Fixed Routing; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

policy performs almost as good as first-fit policy for longer hop lengths. When the traffic load (λ/μ) falls for a given holding time, the effectiveness of the random policy over the first-fit policies become more clear as shown in Figure 5.13 and Figure 5.14

5.3 Fixed (Static) Versus Alternate Routing Schemes

Two different routing schemes were simulated:

- Fixed (Static) routing scheme.
- Alternate routing scheme (with 2 pre-computed paths)

Intuition would lead us to believe that as the number of choices for a route increases, the performance of any wavelength assignment policy and wavelength reservation scheme will get better. We find, however, that this is not necessarily the case. The alternate routing scheme was found to be performing worse than the fixed routing scheme when the holding time was about an order of magnitude less than the diameter of the network. Figure 5.15 through Figure 5.20 indicate this for 5×5-torus and NSFNET networks. Similar trends are

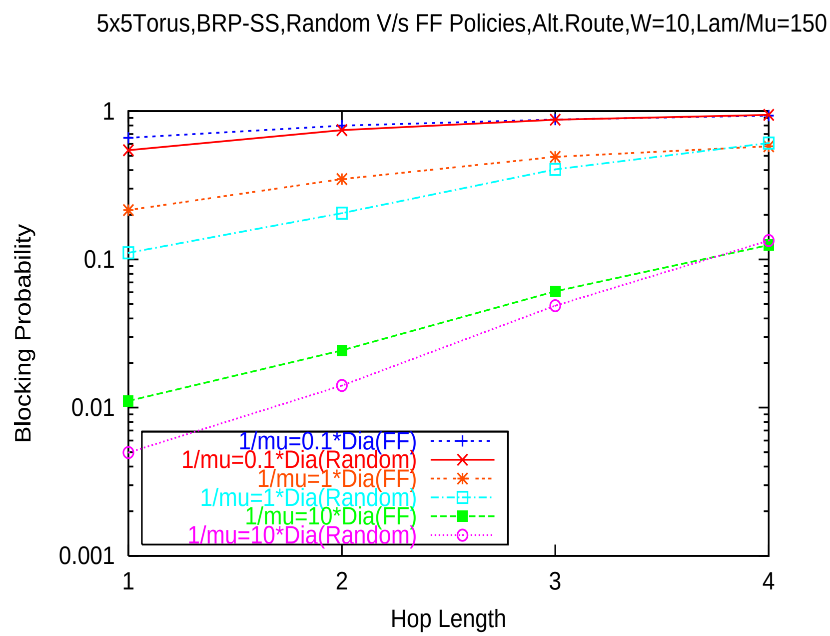

Figure 5.10: Compare Random Versus First-Fit Assignment Policy; BRP Scheme; 5×5-torus; Alternate Routing; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

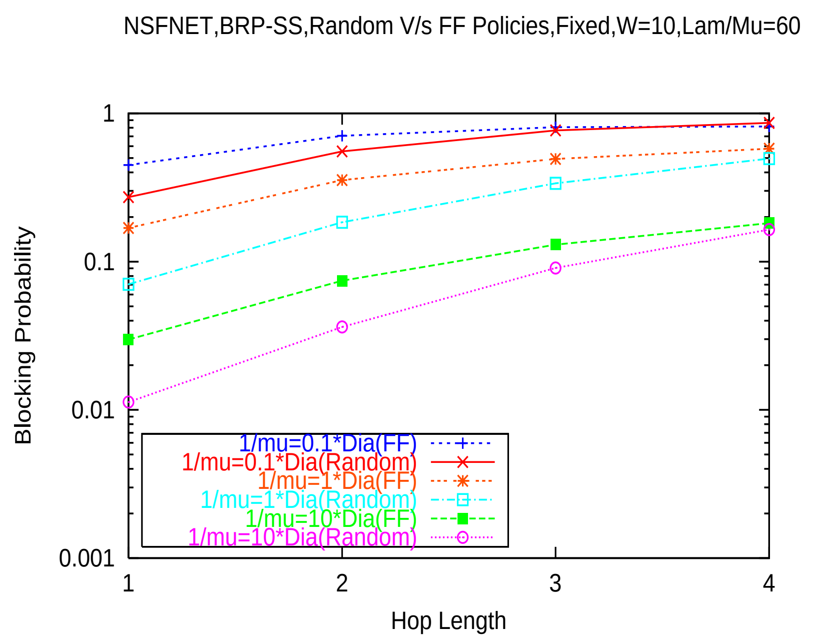

Figure 5.11: Compare Random Versus First-Fit Assignment Policy; BRP Scheme; NSFNET; Fixed Routing; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

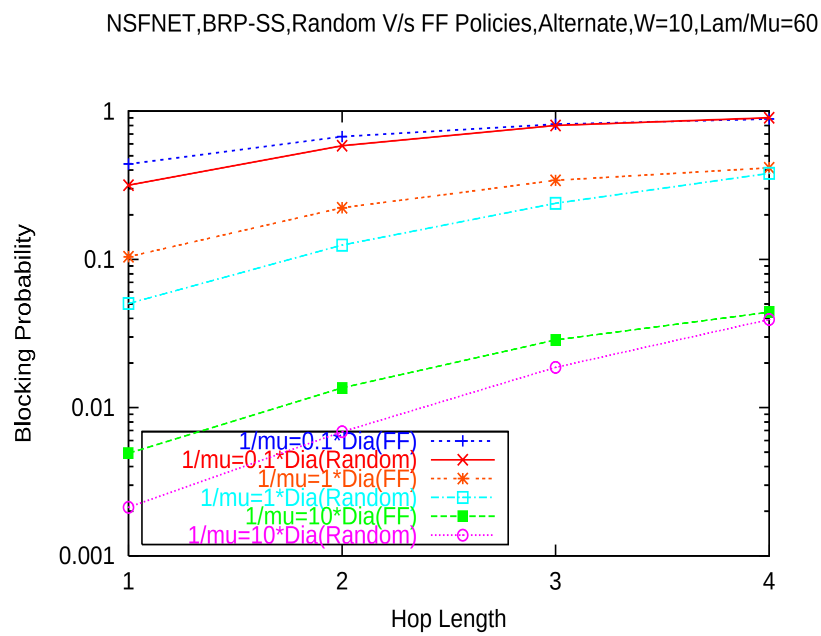

Figure 5.12: Compare Random Versus First-Fit Assignment Policy; BRP Scheme; NSFNET; Alternate Routing; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

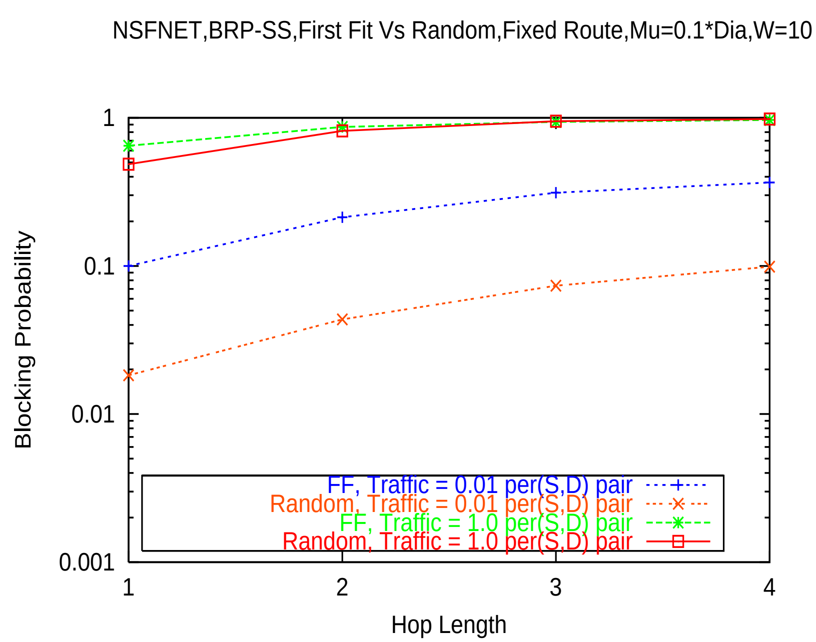

Figure 5.13: Compare Random Versus First-Fit Assignment Policy; BRP Scheme; NSFNET; Fixed Routing; Different Traffic Loads; $1/\mu = 0.1$*Diameter

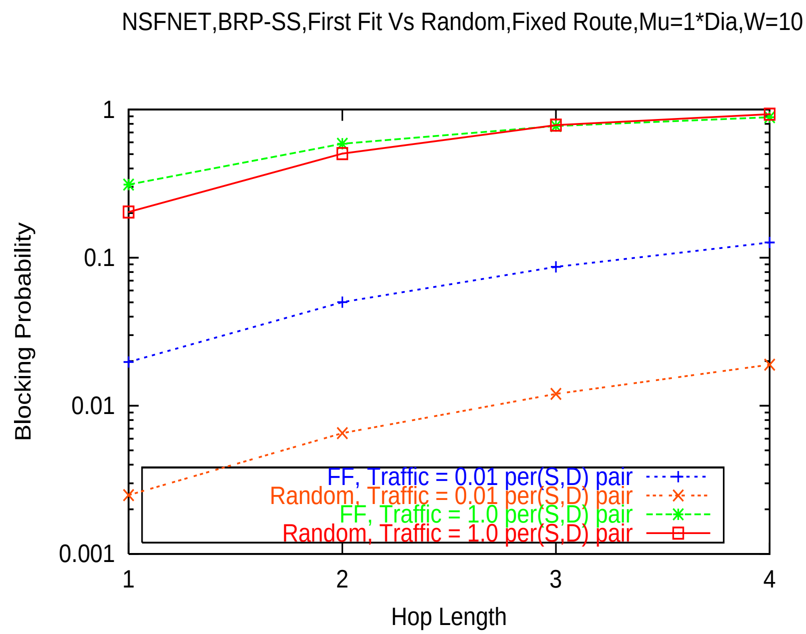

Figure 5.14: Compare Random Versus First-Fit Assignment Policy; BRP Scheme; NSFNET; Fixed Routing; Different Traffic Loads; $1/\mu$ = 1*Diameter

noticed for ring networks but are not shown here. The above stated result may be reasoned out by noticing that in an alternate routing scheme, the resources (wavelengths) tend to get reserved (without being confirmed in the forward direction, as in FRP-SS scheme) or waiting to get released (as in BRP-SS scheme when failure notification goes back to destination node to release the confirmed resources) in far more links and for relatively more time with increasing dominance of signaling delays relative to the holding times. Hence an arriving lightpath request, would find more resources unavailable on an average and hence more blocking results. Traffic load (λ/μ) being constant, with the increase in holding times relative to the signaling delays, the situation in the network gets increasingly stable and the influence of signaling delays on the blocking probability diminishes. Thus, the benefits of alternate routing begin to kick in only as the holding time increases relative to the signaling delyas and may be seen in Figure 5.15 through Figure 5.18 where the the curves for fixed and alternate routing schemes increasingly diverge with increase in holding times for a given hop length. By similar argument, when traffic load increases (network state becomes increasingly dynamic) for a given holding time, alternate routing tends to perform worse than the fixed routing scheme for some cases. Figure 5.19 and Figure 5.20

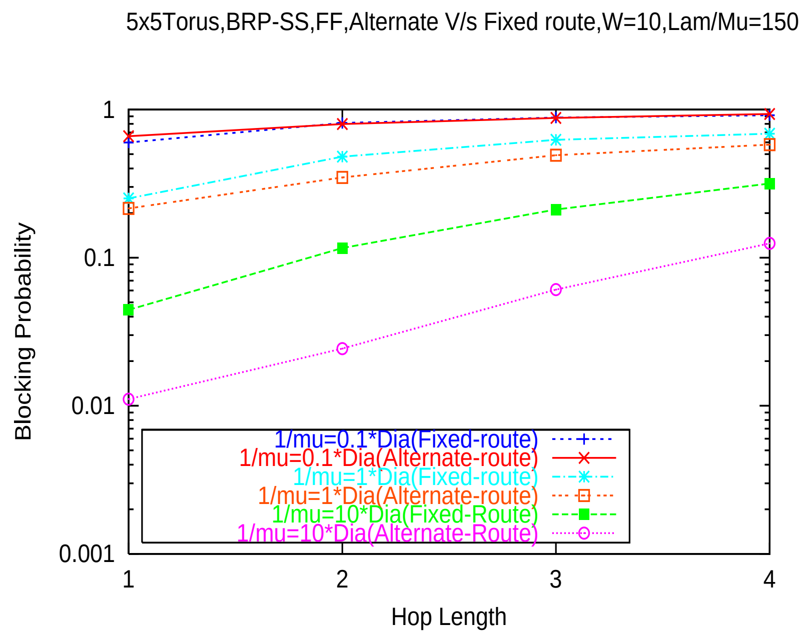

Figure 5.15: Compare Fixed (Static) Versus Alternate Routing Schemes for BRP scheme with First-Fit policy; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

illustrate this for NSFNET topologies. Other trends observed in these graphs are that the blocking probability increases with hop length and the benefits of alternate routing decreases with increasing hop length. As a side result, comparison of Figure 5.15 and Figure 5.17 shows that the performance of first fit and most-used policies are similar, which has been noticed in the past for situations with negligible signaling delays. It is easy to see that first fit and most-used heuristics perform similarly because the most-used heuristic essentially aims to utilizing certain wavelengths more than others and thus a preference of wavelengths gets established which is similar to the pre-defined ordering of wavelengths in the first-fit heuristic.

5.4 BRP-SS Versus FRP-SS

We compare the BRP-SS wavelength reservation scheme with the FRP-SS wavelength reservation scheme. Like earlier results [22], the BRP-SS scheme performs better than the FRP-SS when the first-fit assignment policy is used. The results may be justified by observing that the BRP-SS probes the links in the forward direction without reserving

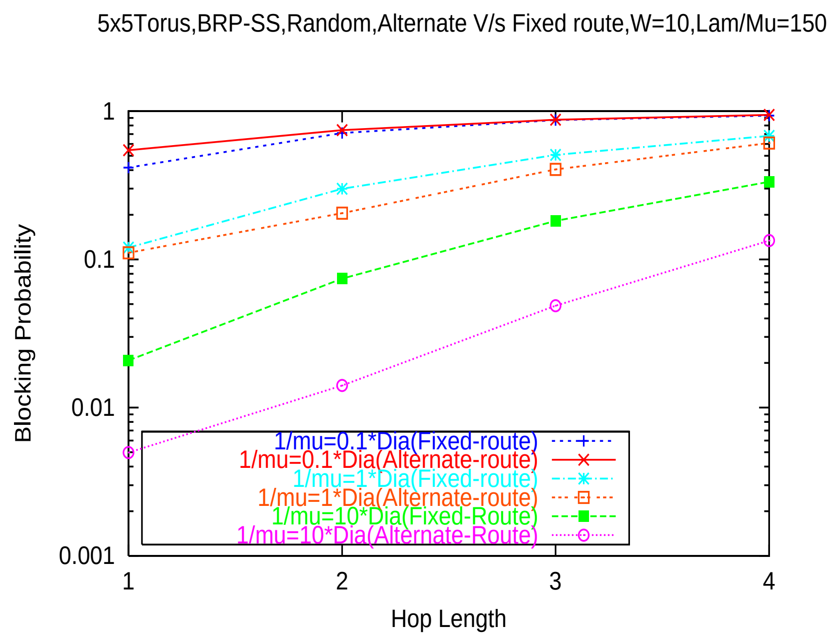

Figure 5.16: Compare Fixed (Static) Versus Alternate Routing Schemes for BRP scheme with Random policy; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

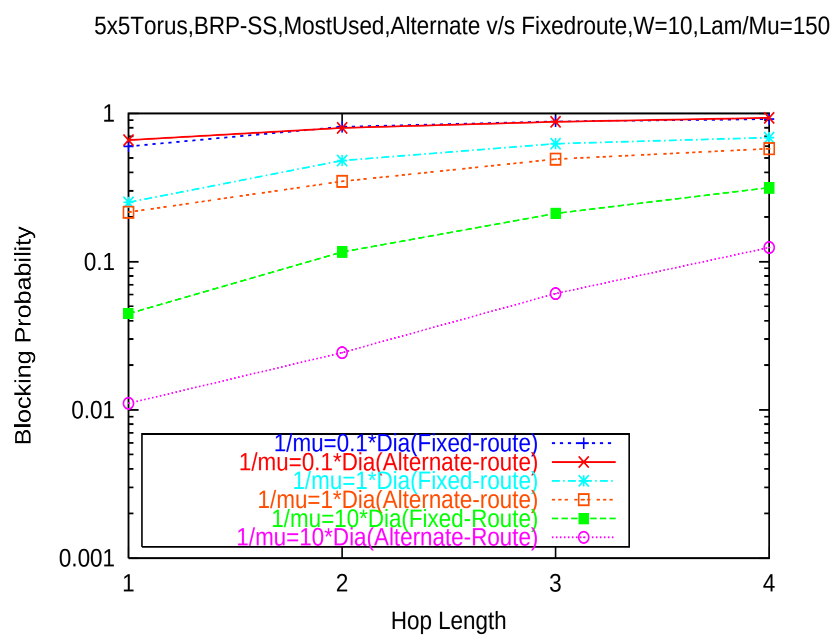

Figure 5.17: Compare Fixed (Static) Versus Alternate Routing Schemes for BRP scheme with Most-Used policy; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

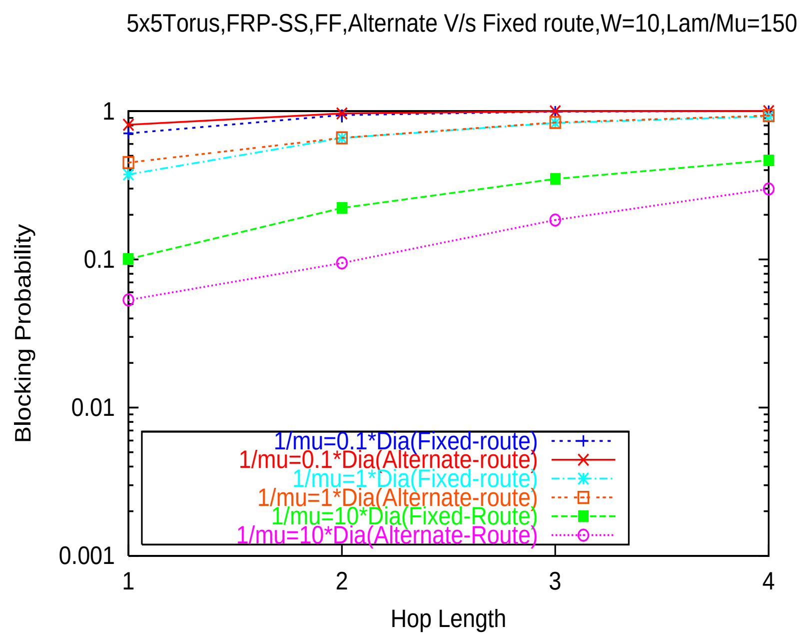

Figure 5.18: Compare Fixed (Static) Versus Alternate Routing Schemes for FRP scheme with First-fit policy; Traffic load (λ/μ) = 0.25 per (S,D) pair

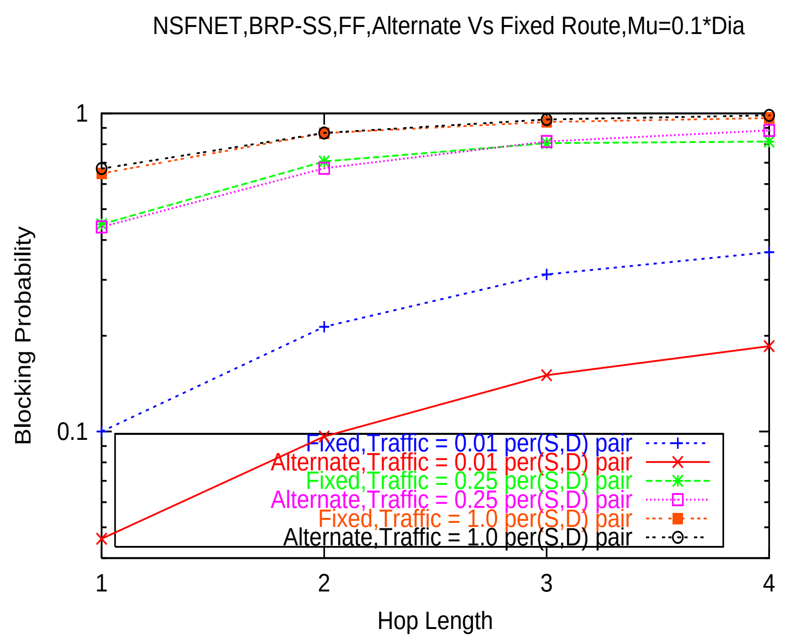

Figure 5.19: Compare Fixed (Static) Versus Alternate Routing Schemes for BRP scheme with First-fit policy; $1/\mu$ = 0.1*Diameter

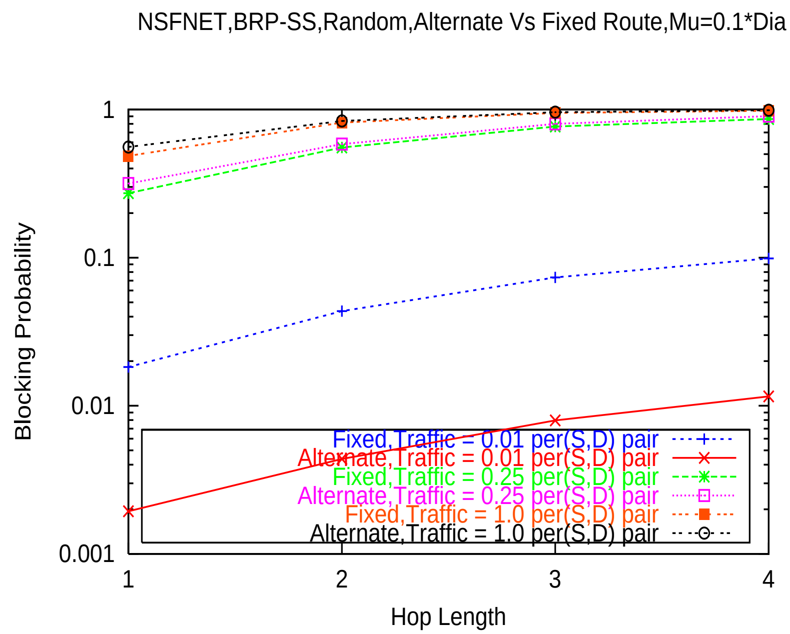

Figure 5.20: Compare Fixed (Static) Versus Alternate Routing Schemes for BRP scheme with Random policy; $1/\mu = 0.1$*Diameter

wavelengths, while the FRP-SS does reserve the wavelengths that are available across all the links in the path thereby 'hoarding' the resources for a far greater time. Hence more blocking results in FRP-SS scheme than BRP-SS scheme. We note that the difference between the performance of the BRP-SS and FRP-SS reservation policies decreases when the traffic load increases or when the holding times decreases relative to the signaling delays, other variables held constant. This is because as the traffic load increases, the contribution to the blocking probability in the forward direction (source to destination) increases in both BRP-SS and FRP-SS schemes, because more active established lightpaths (more resources used up) are present in the network. Thus with increasing traffic, more lightpath requests tend to get rejected right in the probing phase in a BRP-SS scheme which reduces the relative effectiveness of a BRP-SS scheme. Also, with an increase in traffic load, the signaling messages in a BRP-SS scheme face a greater possibility in being rejected while returning to the source node as there is a greater number of lightpath requests trying to establish themselves in the network. Similarly, when the holding time decreases relative to the signaling delays, the network becomes relatively more dynamic leading to increased blocking due to the greater discrepancy in gauging the availability of wavelengths between

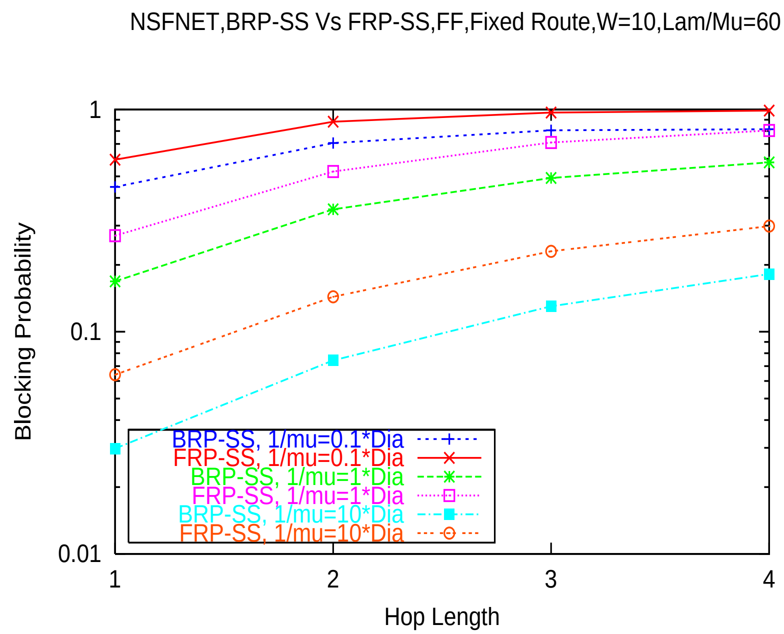

Figure 5.21: Compare BRP-SS scheme with FRP-SS scheme; NSFNET; First-fit policy; Fixed Routing; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

the probing and confirming stages in a BRP-SS scheme. This again, effectively reduces the advantage that BRP-SS scheme has over FRP-SS scheme. The results are shown for NSFNET and 12-node ring topologies in Figure 5.21 through Figure 5.23. Similar trends are observed for 5×5-torus networks but are not shown here.

5.5 Effect of traffic loads(λ/μ)

Increasing traffic load is expected to lead to an increase in blocking probability and is shown in Figure 5.19 and Figure 5.20. When traffic load increases, there are more lightpath requests that arrive which compete for the same set of resources and hence blocking increases. This is an expected and well-known result and is documented here only for completion.

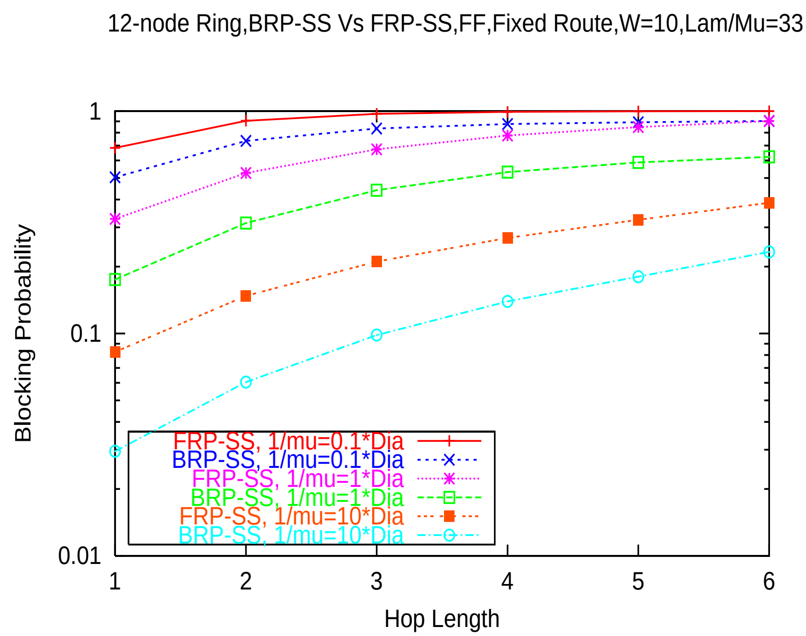

Figure 5.22: Compare BRP-SS scheme with FRP-SS scheme; 12-node Ring; Fixed Route; First-fit policy; Traffic load $(\lambda/\mu) = 0.25$ per (S,D) pair

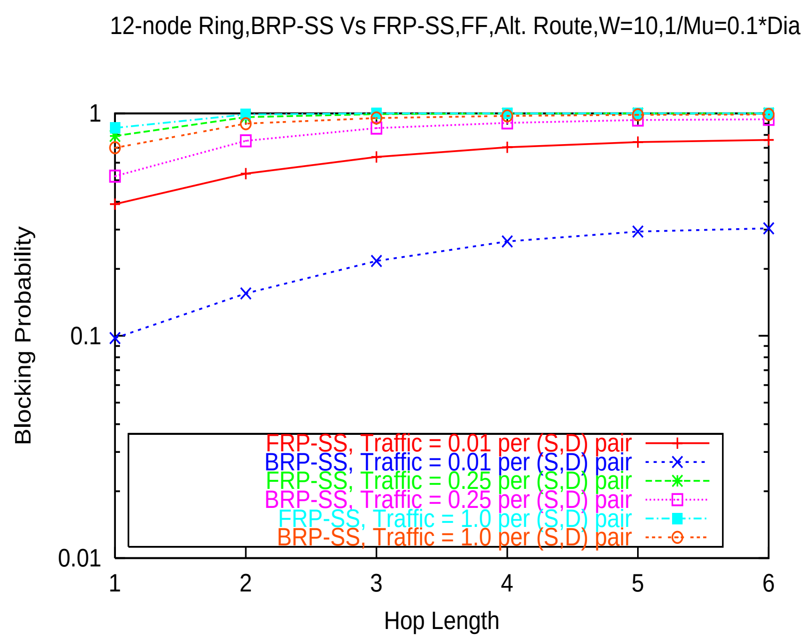

Figure 5.23: Compare BRP-SS scheme with FRP-SS scheme; 12-node Ring; Alternate Route; First-fit policy; $1/\mu = 0.1$*Diameter

Chapter 6

Summary & Future Work

6.1 Summary

We studied the performance of various wavelength assignment policies and wavelength reservation techniques in wavelength-routed second-generation optical wide area networks (WANs) while taking into account the effects of signaling delays. We developed a simulator using the C++ language to study the above as we desired to simulate the signaling protocol using the semantics of CR-LDP, a relatively new signaling protocol proposed by the mpls working group of IETF. Although the effects of signaling delays have been studied in the past ([18], [22]), we believe that a work, such as ours, taking into account the combined effects of holding time relative to the signaling delays, traffic load, wavelength assignment policy, routing scheme, and wavelength reservation scheme for both regular and irregular topologies has not been attempted before.

We got a couple of interesting results and a few which reinforced conventional knowledge. A couple of expected results were that blocking probability increased with increasing traffic load, and decreasing holding times. We also could show that the BRP-SS wavelength reservation scheme performed better than the FRP-SS reservation scheme when considered along with the first-fit wavelength assignment policy. This was observed for various topologies, routing schemes and traffic loads. Previous study [22] indicates that the BRP schemes in general, outperform , FRP schemes. One interesting result was that random wavelength assignment policy performs better (less blocking probability) than the first-fit assignment policy and the difference in performance between the two reduces with increasing hop length. We provided an intuitive explanation of the result. This result has not been published anywhere yet to the best of our knowledge and belief. First-fit policy, till now, was generally accepted to perform better than random policies because studies in the past have generally assumed zero signaling delays. Another interesting result is that alternative routing scheme need not always perform better than the fixed routing scheme. We showed that when holding times are an order of magnitude lower when compared to the diameter of the network, alternative routing scheme may actually perform worse than the fixed routing scheme. We provided an intuitive explanation for this also. We have reasons to believe that this result also has not been published yet.

6.2 Future Work

Our work may be extended in many ways some of which we outline. Since the effect of non-negligible signaling delays led to some interesting results, we believe it is perhaps worthwhile to build components for 'ns' simulator so that the semantics of signaling protocols like CR-LDP or RSVP-TE can be simulated. Efforts in this direction have been taken where the design objectives of such a simulator [3], architecture and implementation details [21] are provided. While the simulator described therein has the ability to specify important elements such as limited or full wavelength conversion in a node, there is no mention of any wavelength reservation scheme or the signaling delays being incorporated in their implementation.

Another possible area of research is to analyse this problem from an analytical view point. While most flavors of RWA problem are analytically difficult to solve, it may be interesting and useful to develop an approximate analytical method for the above problem.

The features of CR-LDP like pre-emption of lightpaths, the impact of full or limited wavelength conversion at nodes, were not simulated in our study and could be a possible area of extending this work.

Bibliography

[1] ANDERSSON, L., DOOLAN, P., FELDMAN, N., FREDETTE, A., AND THOMAS, B. LDP Specification, January 2001. Available from URL http://www.ietf.org/rfc/rfc3036.txt?number=3036.

[2] BARRY, R. A., AND HUMBLET, P. A. Models of blocking probability in all-optical networks with and without wavelength changers. *IEEE Journal on Selected Areas in Communications 14*, 5 (June 1996), 858–867.

[3] BHIDE, N. M., AND SIVALINGAM, K. M. Design of OWns: Optical wavelength division multiplexing (WDM) network simulator, January/February 2000. Available from URL http://catss.utdallas.edu/ONW2000/proceedings/p11.pdf.

[4] BIRMAN, A. Computing approximate blocking probabilities for a class of all-optical networks. *IEEE Journal on Selected Areas in Communications 14*, 5 (June 1996), 852–857.

[5] BORELLA, M. S., P.JUE, J., BANERJEE, D., RAMAMURTHY, B., AND MUKHERJEE, B. Optical components for WDM lightwave networks. *Proceedings of the IEEE 85*, 8 (August 1997), 1274–1306.

[6] CHLAMTAC, I., GANZ, A., AND KARMI, G. Lightpath communications: An approach to high-bandwidth optical WAN's. *IEEE Transactions on Communications 40* (July 1992.), 1171–1182.

[7] FABRY-ASZTALOS, T., BHIDE, N. M., AND SIVALINGAM, K. M. Adaptive weight functions for shortest path routing algorithms for multi wavelength optical WDM net-

works. In *Proceeding of IEEE International Conference on Communications, (New Orleans, LA)* (June 2000), vol. 3, pp. 1330 – 1334.

[8] Harai, H., Murata, M., , and Miyahara, H. Performance of alternate routing methods in all-optical switching networks. In *Proceedings of IEEE INFOCOM* (1997), vol. 2, pp. 517–525.

[9] Jamoussi, B., Andersson, L., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N., Fredette, A., Girish, M., Gray, E., Heinanen, J., Kilty, T., and Malis, A. CR-LSP setup using LDP, January 2002. Available from URL `http://www.ietf.org/rfc/rfc3212.txt?number=3212`.

[10] Kovacevic, M., and Acampora, A. Benefits of wavelength translation in all-optical clear-channel networks. *IEEE Journal on Selected Areas in Communication 14* (June 1996), 868–880.

[11] Li, L., and Somani, A. A new analytical model for multifiber WDM networks. *IEEE Journal on Selected Areas in Communications 18*, 10 (October 2000), 2138–2145.

[12] Li, L., and Somani, A. K. Dynamic wavelength routing using congestion and neighborhood information. *IEEE/ACM Transactions on Networking (TON) 7*, 5 (October 1999), 779–786.

[13] Mokhtar, A., and Azizoglu, M. Adaptive wavelength routing in all-optical networks. *IEEE/ACM Transactions on Networking (TON) 6* (April 1998), 197–206.

[14] Ramamurthy, S., and Mukherjee, B. Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks. In *Proceedings of IEEE GLOBECOM (Sydney, Australia)* (1998), pp. 2295–2303.

[15] Ramesh, S., Rouskas, G. N., and Perros, H. G. Computing call blocking probabilities in multi-class wavelength routed networks. *ACM Transactions on Modeling and Computer Simulation 10*, 2 (April 2001), 87–103.

[16] R.Dutta, and Rouskas, G. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks Magazine* (January 2000).

[17] Rosen, E., Viswanathan, A., and Callon, R. Multiprotocol label switching architecture, January 2001. Available from URL http://www.ietf.org/rfc/rfc3031.txt?number=3031.

[18] Saha, D. A comparative study of distributed protocols for wavelength reservation in WDM optical networks. *Optical Networks Magazine 3*, 1 (January/February 2002), 45–52.

[19] Sivalingam, K. M., and Subramaniam, S. *OPTICAL WDM NETWORKS: Principles and Practice*. Kluver Academic Publishers, 2000.

[20] Subramaniam, S., Azizoglu, M., and Somani, A. A performance model for wavelength conversion with non-poisson traffic. In *Proceedings of IEEE INFOCOM* (April 1997), pp. 500–507.

[21] Wen, B., Bhide, N. M., Shenai, R. K., and Sivalingam, K. M. Optical wavelength division multiplexing (WDM) network simulator (OWns): Architecture and performance studies. *Optical Networks Magazine Special Issue on 'Simulation, CAD, and Measurement of Optical Networks'* (March 2001). Available from URL http://www.eecs.wsu.edu/~dawn/Papers/2001/ONM01b.pdf.

[22] X.Yuan, R.Melhem, R.Gupta, Y.Mei, and C.Qiao. Distributed control protocols for wavelength reservation and their performance evaluation. *Photonic Networks and Communications 1*, 3 (1999), 207–218.

[23] Zang, H., Jue, J., and Mukherjee, B. A review of routing and wavelength assignment approaches for wavelength routed optical networks. *Optical Networks 1*, 1 (January 2000), 47–60.

[24] Zhu, Y., Rouskas, G. N., and Perros, H. G. A comparison of allocation policies in wavelength routed networks. *Photonic Network Communications 2*, 3 (August 2000), 265–293.

[25] Zhu, Y., Rouskas, G. N., and Perros, H. G. A path decomposition approach for computing blocking probabilities in wavelength routed networks. *IEEE/ACM Transactions on Networking (TON) 8*, 6 (December 2000), 747–762.