2002

A generalized framework for analyzing time-space switched optical networks

Srinivasan Ramasubramanian
Iowa State University

A generalized framework for analyzing time-space switched optical networks

by

Srinivasan Ramasubramanian

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Arun K. Somani, Major Professor
Manimaran Govindarasu
Ahmed E. Kamal
Sunder Sethuraman
Robert J. Weber

Iowa State University

Ames, Iowa

2002

Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of

Srinivasan Ramasubramanian

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

DEDICATION

To my parents, Rukmani and Ramasubramanian,

for their trust, support, encouragement, and love.

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Prof. Arun K. Somani, for his guidance and encouragement throughout my graduate study at Iowa State University. His professional advice and insights have always led me to the right directions in research and career. He provided an excellent learning atmosphere that not only allowed me to undertake research in optical networks, but also in other areas in Computer Engineering as well. I have always been in awe seeing him work. His half-an-hour meetings with his students in the mornings have always amazed me. He discusses Optical Networks, Wireless Networks, Computer Architecture, Fault tolerance, and Reconfigurable Computing in those two hours in the morning. He needs less than a minute to switch to a completely different topic and talk to his students in their wavelength. He has always been and will remain my role model in my academic endeavors. I certainly hope that we continue to work together in future.

I would like to thank my committee members, Prof. Robert J. Weber, Prof. Ahmed E. Kamal, Prof. Manimaran Govindarasu, and Prof. Sunder Sethuraman, for their valuable suggestions and comments to improve the quality of the dissertation. Special thanks to Prof. Ahmed E. Kamal for reading my dissertation meticulously and offering insightful comments. Also, special thanks to Prof. Manimaran Govindarasu for all his friendly talks that were packed with spirit and encouragement, especially on days when I heard my paper got rejected or when I was stuck with a problem and cannot find my way out.

Thanks to my fellow labmates for their companionship that made my research life a memorable one: Govindarajan Krishnamurthi, Ling Li, Seongwoo Kim, Huesung Kim, Sashisekaran Thiagarajan, Murari Sridharan, Deepali Deshpande, Rama Sangireddy, Raed Al-Omari, Jing Fang,

Pallab Datta, Anirban Chakrabarthi, Rakesh Raghavan, and Aaron Striegel. Thanks to Nagaraj Balijepalli – for all the wonderful coffee-room discussions on all matters of life; Naveen Vijayaraghavan, Subramanian Lalgudi, Swaminathan Sivasubramanian, Vineela Manne, and Vaishnavi Madineni – for all the fun times during the last year; Murali Viswanathan – for giving me tough competition in the racquetball court and help me improve my game; Sashisekaran Thiagarajan and Murari Sridharan – for all the good times that we had in the last five years.

Thanks to Vinetha Narasimhan, for always being there (Yes, for being out there in Canada. Just kidding.). I have always looked forward to talking to her, especially after those stressful days in lab. I cannot thank her enough for giving a life to my nerdy school days. I have enjoyed every bit of our talks and I appreciate her for putting up with all my academic babble. She has been a truly wonderful friend.

Ames has seen me through a very important phase of my life – from a naive fresh graduate student to a mature doctoral degree holder (Oh yes! We have seen Ames grow up too – from a small campus town to attain the Metropolitan status.). I have had many friends during the course of my five years of stay in Ames. They have all influenced my life in one way or the other. It is hard to list all their names here and the impact they had on my life, but they will always be remembered as an indispensable part of my life in Ames.

I sincerely appreciate the valuable help of Nancy Knight, Graduate Co-ordinator, and Pam Myers, Graduate Secretary, in every step of the academic process. They had answers to every question of mine and made all the paper work look simple. Thanks to the Computer Support Group (CSG) of the Electrical and Computer Engineering Department for their timely and excellent computational support. A big thank-you to the Department of Electrical and Computer Engineering at Iowa State University for their excellent resources, infrastructure, and just about anything else. The coffee room in Coover (Take my word, most faculty do not know where it is.) has hosted several interesting discussions and has been the birth-place for most of our research ideas. Every graduate student knows that if he/she cannot find someone in the lab, the next obvious location is the coffee room. I hope it is retained in its true form in the years to come.

Just words cannot express my deep sense of gratitude to my parents, Rukmani and Ramasubramanian, and my brother Ramamoorthy (Ramesh). They have been, and will always remain, the first people that I look forward to for sharing my successes and happiness with. They have cherished my successes even more than I did myself. If it is not for their sacrifices and hardship, I could not not have written this dissertation.

Finally, my sincere thanks to all the teachers I have had in my entire student life. Irrespective of the subject they taught, every one of them have played a vital role in getting me to where I am today. They will always be remembered for their efforts that provided a strong foundation over which I have built my academic career.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Wavelength division multiplexed (WDM) optical networks have emerged as the viable solution to providing high-speed networking solutions that meet the increasing demands of the Internet. WDM networks divide the fiber bandwidth into multiple WDM channels called *wavelengths*. The current transmission speed on a wavelength is 10 Gigabits per second and is expected to increase to 40 Gigabits per second in the near future. The user requirement, on the other hand, are of sub-wavelength capacity requirement ranging from 155 Megabits per second to a maximum of 2.5 Gigabits per second. Such a mis-match between the user requirements and the minimum granularity of a connection offered by the network has necessitated the use of efficient wavelength sharing mechanisms. Networks that allow multiple users to share the wavelength bandwidth are referred to as WDM grooming networks.

In this dissertation, we propose a novel conceptual network model called *Trunk Switched Network* (TSN) to model WDM grooming networks. The links in a network are viewed as a set of channels. The channels are combined at a node to form groups called *trunks*. The nodes can switch channels across different links that fall within the same trunk. We model WDM grooming networks with different grooming capabilities as TSNs. We develop a framework for connection establishment in TSNs from which different path selection and channel assignment schemes can be derived. We analyze a TSN for blocking performance. The analytical model based on fixed-path routing mechanism includes three main features: (1) incorporates heterogeneous switching architectures; (2) considers multi-rate traffic streams; and (3) employed to evaluate blocking performance of multicast tree establishment, which is a generalized form of a path. We also study the effect of dynamic routing in WDM grooming networks that employ wavelength-level traffic

grooming. We propose a request-specific routing algorithm called Available Shortest Path (ASP) routing that aims at minimizing the network resources used by a connection. In this approach, a path for establishing a connection is chosen based on the request characteristics. We employ dispersity routing that splits a request with larger capacity requirement into multiple requests with smaller capacity requirement. We show that request-specific routing and dispersity routing can be employed as a cost-effective alternative to improving network performance as compared to the high-cost solution of increasing grooming capability at the nodes in the network.

The network model proposed in this dissertation is a conceptual framework and does not depend on technological aspects. Hence, it can be applied in general to networks that employ multiplexing in more than one domain, such as time, frequency/wavelength, code, phase, etc. and employ switching within a sub-set of the channels. We believe that the proposed network model and the results discussed in this dissertation will have significant impact in the design and operation of future high-speed backbone networks.

CHAPTER 1 Introduction

Technological advances in the area of telecommunications over the last half a century has finally enabled the communication networks see the *light*. Current telecommunication transmission lines employ light signals to carry information over guided channels called *optical fibers*. The transmission of signals that travel at speed of light is not new and has been in existence in the form of radio broadcast for the last several decades. However, such a transmission technology over a guided medium, unlike air, with very low attenuation and bit error rate makes the optical fibers a natural choice for medium of communication for the next-generation high-speed networks.

Staring in the early 20th century, telecommunications has grown steadily with the advances in electronics. Technological advances in semiconductor products fueled this improvement simplifying the long-distance communication infrastructure. With the advent of optical transmission technology over optical fibers, the communication networks have attained orders of magnitude increase in the network capacity. The bandwidth available on a fiber is approximately 25 terahertz. In late 1980's and early 1990's, the migration from electronic networks to optical transmission technology involved only replacing copper cables with optical fibers. Traditional time division multiplexing (TDM) that allows multiple users to share the bandwidth of a link was employed. In TDM, the bandwidth sharing is in the time domain. Multiplexing techniques specific to optical transmission technology were not employed in the early networks. Synchronous Optical NETwork (SONET) is the most popular network in this category [1]. SONET is based on a ring-architecture employing circuit-switched connections to carry voice traffic.

Increasing the transmission speed could not be adopted as the only means of increasing the network capacity. Transmission speeds beyond a few tens of gigabits per second could be be sustained for longer distances for reasons of impairments due to amplifiers, dispersion, non-linear effects of fiber, and cross-talk [2]. Hence, wavelength division multiplexing (WDM) was introduced that divided the available fiber bandwidth into multiple smaller bandwidth units called *wavelengths*. Different connections can share the bandwidth on a link using different wavelengths. Such a multiplexing mechanism divides the bandwidth *space* into smaller portions. Hence, the multiplexing is said to occur in the space domain.

1.1 Broadcast-and-select networks

Early optical networks employing WDM were broadcast and select in nature. In such networks, the information that is transmitted by a node is received by all other nodes. Every node takes the information that is destined for it and discards the rest. Figure 1.1 shows a broadcast and select network. Nodes in these networks are connected by links and optical couplers. An optical coupler is a *passive* component that is used for either combining or splitting the signals into or from a fiber. The couplers can be configured to split/combine signals in specific ratios. Nodes in these networks transmit information on specific wavelengths. Typically, a network with N nodes employ N wavelengths where every node is uniquely assigned a wavelength. Every node in the network receives one transmitter and N receivers. If at any time information on only one channel is needed, then one tunable receiver is employed at every node, instead of N dedicated receivers. In networks where the number of nodes is greater than the number of wavelengths per fiber, wavelengths are shared among multiple nodes. Such a scheme would then require the need for channel access protocols to share the wavelength.

The major features of optical couplers include low insertion loss, excellent environmental stability, long-term reliability, and multiple performance levels. Such networks are typically found in local area networks and Cable-TV or video distribution networks, networks that typically provide the end-user connectivity.

The disadvantage of such a passive network is that its range is limited. Long range networks, typically country-wide networks, cannot employ such a broadcast and select mechanism due to

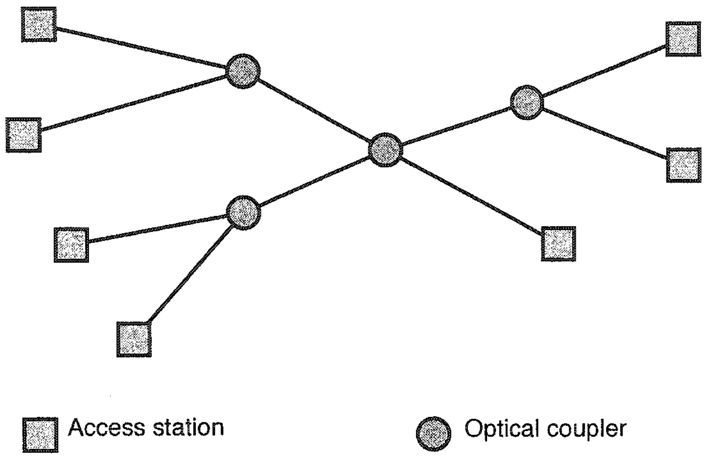


Figure 1.1 A broadcast and select network.

capacity inefficiency. The information is unnecessarily sent to all the nodes in the network resulting in a poor network utilization. Also, as the signals travel farther the quality of the signal degrades necessitating signal re-generation with re-shaping and re-timing.

1.2 Wavelength-routed networks

In order to avoid unnecessary transmission of signals to nodes that do not require them, *wavelength-routing* was introduced. In wavelength-routed networks the nodes employ *optical cross-connects* that can switch an individual wavelength from one link to another. In order to operate the network in a transparent manner, the switching of a wavelength is done in the optical domain.

A connection from one node to another established on wavelength is referred to as a *lightpath* [3]. A Wavelength-routed WDM network is shown in Figure 1.2. The figure shows connections established between nodes 1 and 5, 2 and 5, and 3 and 6. The connections from nodes 1 and 2 share two links. Hence, they have to occupy different wavelengths on those links.

Wavelength conversion is a mechanism by which an optical signal from one wavelength is converted into another. A device that performs such a conversion is referred to as *wavelength converter*. Wavelength conversion mechanisms can be classified based on the range of wavelength

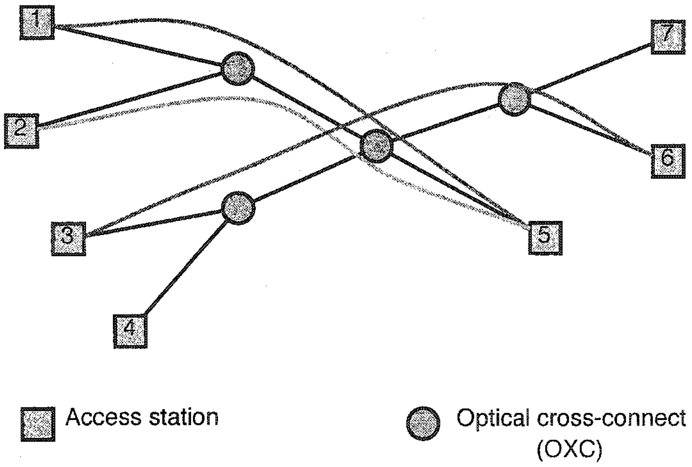


Figure 1.2 A wavelength-routed WDM network.

conversion. *Fixed-wavelength conversion* allows the signal to be converted from a specific input wavelength to a fixed output wavelength. The choice of output wavelength is fixed for an input wavelength, hence the name. If the signal on a wavelength can be converted into any other wavelength, it is referred to as *full-wavelength conversion*. If the signal can be converted from one wavelength to a set of but not all wavelengths, it is referred to as *limited-wavelength conversion*. Figs. 1.3 shows the different types of wavelength conversions for a network with four wavelengths denoted by $W1$ through $W4$.

The architecture of a node in a wavelength-routed WDM network that does not employ wavelength conversion is shown in Figure 1.4. The signals from different wavelengths on a link are first de-multiplexed at a node. The individual wavelengths from the different links are then fed into a wavelength switch that switches the wavelengths across different links.

The architecture of a node employing full-wavelength conversion is shown in Figure 1.5. The figure shows a node with four links with each link having three wavelengths. After de-multiplexing the wavelengths from the links, the individual wavelengths are fed into tunable wavelength converters that convert the signals to the corresponding output wavelength. Every wavelength on every link is provided a dedicated wavelength converter. While such an architecture allows any wave-

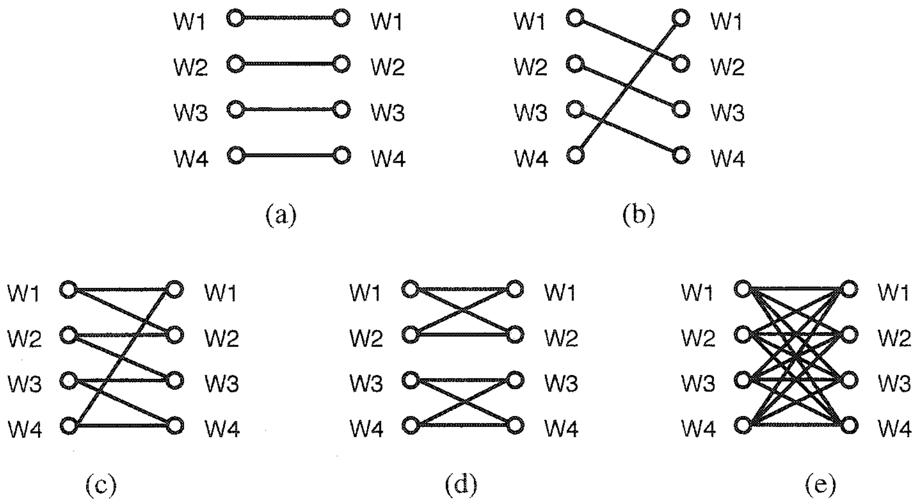


Figure 1.3 Types of wavelength conversion. (a) No wavelength conversion; (b) Fixed-wavelength coversion; (c,d) Limited-wavelength conversion; (e) Full-wavelength conversion.

length to be converted to any other wavelength at the node, most of the connections could be routed without the need for wavelength converters. Hence, such an approach could result in wastage of wavelength converter resources. Note that wavelength converters are expensive components, hence cannot be employed at a node at large numbers unless there is a strong need for it.

Figure. 1.6 shows the architecture of a node that employs limited number of wavelength converters that can be shared by connections on different links. The set of wavelength converters are referred to as *wavelength converter bank*. In this case, there are four converters available in the bank. The output of the wavelength converters are connected to a switch that could have more output ports than input ports. Such an architecture allows for more than one wavelength converter to be used connections on an output link. In the architecture shown in Figure. 1.6 up to four connections can avail the facility of wavelength conversion at the node with up to two connections on an output link employing wavelength converters. If the wavelength converters are not needed, then the connections are switched directly at the first switch to the respective output links.

The wavelength converter blocks that are shown in the node architectures can be implemented

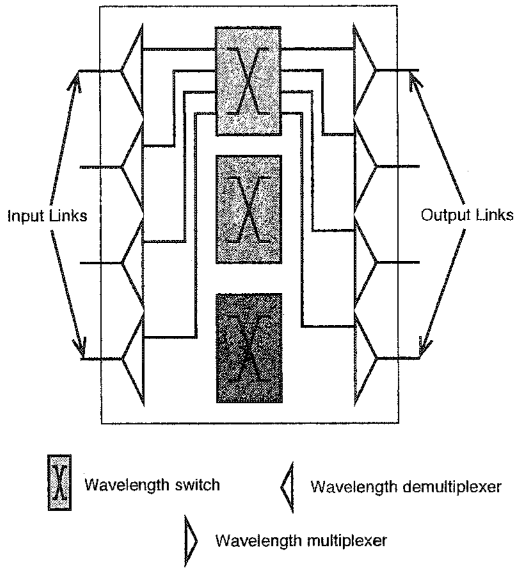


Figure 1.4 Node architecture in a wavelength-routed WDM network without wavelength conversion.

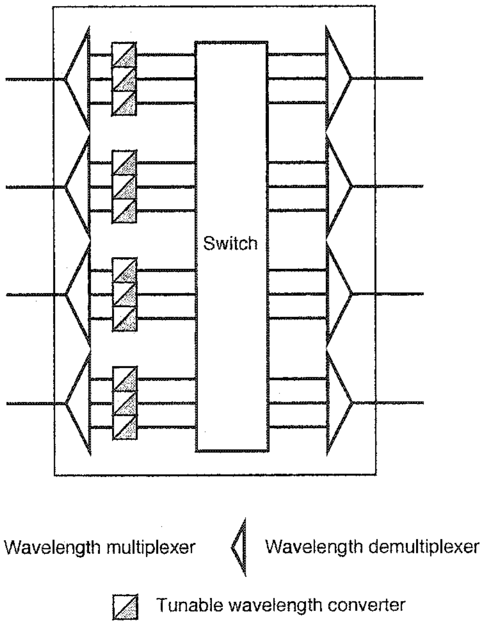


Figure 1.5 Node architecture in a wavelength-routed WDM network with wavelength conversion at input ports.

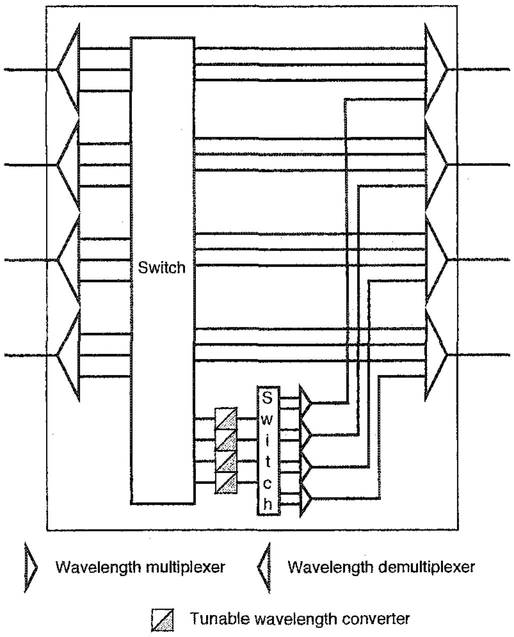


Figure 1.6 Node architecture in a wavelength-routed WDM network with wavelength conversion bank at output ports.

in several ways. A trivial yet practical solution is to receive the incoming signal in the electronic domain and re-transmit in the desired output wavelength. However, such an approach compromises on the transparency of the network. The clock-rate of the incoming signal and frame format must be known to the intermediate nodes that employ wavelength conversion. Such a lack of transparency also limits the scalability of the network.

All-optical wavelength conversion retains the signal in the optical domain and is transparent to the clock-rate and frame format. However, these devices are prohibitively expensive to be deployed widely in the networks.

1.3 Multi-fiber networks

The expensive proposition of employing wavelength converters forced researchers to venture into alternatives for wavelength conversion. Multi-fiber networks is an alternative to employing wavelength conversion.

Multi-fiber networks employ more than one fiber on a link between two nodes. Hence, every link in the network has multiple fibers employing multiple wavelengths. The advantage of having such a network is that a wavelength on a link from an input fiber can be switched to any of the fibers on the output link. Figure. 1.7 shows the switching possibilities of different wavelengths at a node that does not employ wavelength conversion. Each link is assumed to have two fibers (F1 and F2) and two wavelengths (W1 and W2) per fiber. Such a multi-fiber approach is similar to that of limited-wavelength conversion capability, refer to Figure 1.3(d).

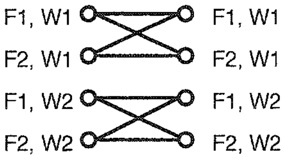


Figure 1.7 Switching possibilities in a multi-fiber network employing two fibers (F1 and F2) and two wavelengths per fiber (W1 and W2).

Employing multiple fibers in a network has a key advantage. As compared to a single-fiber network have the same link capacity, the number of wavelengths in a fiber is reduced by a factor of number of fibers employed. Lesser number of wavelengths per fiber implies that the spacing between two wavelengths can be increased resulting in the use of simpler and less expensive components. Such an approach allows one to increase the signal power in a wavelength, thereby increasing the signal to noise ratio, and requires lesser amplification, resulting in an increased network span. Components requiring to function in a narrow bandwidth are more expensive as compared to those that work in a wider bandwidth. However, the down side of employing multiple fibers is that every fiber needs its own amplifier. Improvements in transmission and fiber technology have made it possible to transmit signals over longer distances without the need for amplification. With such improvements in technology, multi-fiber approach seems to be an attractive alternative for wavelength conversion.

1.4 WDM grooming networks

The minimum granularity of a connection in a wavelength-routed network is the capacity of a wavelength. The transmission speed on a wavelength increases with the advances in the transmission technology. However, the requirement of end-users such as Internet Service Providers (ISPs), Universities, and industries are still much lower than that of the capacity of a wavelength. For example, the connection between Iowa State University and Iowa Communiations Network (ICN) has a bandwidth of 155 Mbps, referred to as OC-3 connection. The bandwidth requirement is projected to increase in future, however even doubling the current bandwidth would be more than sufficient to handle the projected demand for the near future. The current transmission speed on a wavelength are 10 Gbps (OC-192). As of this writing, 40 Gbps (OC-768) technology is commercially available, however is not widely deployed.

The large gap between the user requirement and capacity of a wavelength has forced the need for wavelength sharing mechanisms that would allow more then one user to share a wavelength. Wavelength sharing, similar to sharing a fiber using multiple wavelengths, can be done in several ways. One approach to sharing a wavelength is to divide the wavelength bandwidth into frames containing a certain number of time slots. A time slot on successive frames would then form a channel. Other approaches such as phase modulation and optical code division multiple access (OCDMA) can also be employed to shared the capacity on a wavelength. In this dissertation, we will assume that the wavelength sharing is achieved in the time domain, however, the scope of dissertation is open to all wavelength sharing domains.

The merging of traffic from different source-destination pairs is called *traffic grooming*. Nodes that can groom traffic are capable of multiplexing/de-multiplexing lower rate traffic onto a wavelength and switching them from one lightpath to another. The grooming of traffic can be either static or dynamic. In static traffic grooming, the source-destination pairs whose requirements will be combined are pre-determined. In dynamic traffic grooming, connection requests from different source-destination pairs are combined depending on the existing lightpaths at the time of request arrival.

Recent advances in optical switching technology, as in [4, 5] and [6], have shown the possibility of realizing fast all-optical switches with switching time less than a nanosecond. The use of such fast switches along with fiber delay lines as time-slot interchangers [7, 8] have opened up the possibility to realize multi-fiber, multi-wavelength, optical time switched networks. These networks will be referred to as *WDM-TDM Switched* or *WDM Grooming* networks. WDM grooming networks employing WDM and TDM employ multiplexing mechanisms both in space and time domain.

Converting the signal from one time slot on a wavelength to another is achieved by a device called *time slot interchanger* (TSI). As the signals are transmitted using photons, that do not have valency, it is not possible to store the energy. In other words, there is no equivalent of a memory in the optical domain as that available in the electronic domain. The optical version of TSIs are designed using *fiber delay lines* that are loops of fibers through which the signal is passed from one end and received at the other. Depending on the length of the fiber, different delay times can be achieved. The length of a fiber needed to delay a signal by one time slot interval is the product of the time unit and the speed of light. Delaying by more than one time slot can be achieved by cascading multiple delay units. Programmable delay lines can be implemented with switches and fiber delay lines by selectively activating specific stages of delay units.

We now define a WDM grooming network more formally here. A WDM grooming network consists of switching nodes interconnected by links. A link ℓ has F fibers with each fiber carrying W wavelengths. Every wavelength is divided into frames with T time slots per frame. A *channel* is defined as a specific time slot on successive frames. The link has a total of FWT channels in it. Every channel on a link is denoted by a four-tuple (l, f, w, t) that denotes the link, fiber, wavelength, and time slot identifier on the link.

Requests or *calls* arrive in network that require a certain number of channels from a source to destination node. The requests are accepted in the network along a certain path by assigning the requested number of channels on each link such that every intermediate node on the link can switch a specific input time slot to a specific output time slot. The switching capability of at a node restricts the output channels to which an input channel can be switched to. For example, if a node does not employ wavelength conversion and time slot interchange, then an input channel $(l, f, w, t)_i$ can be switched to $(l, f, w, t)_o$ if and only if $w_i = w_o$ and $t_i = t_o$. The restriction on being the same wavelength is removed if wavelength converters are available. Similarly, TSIs remove the constraint on the same time slot to be maintained at the input and output.

Different nodes in the network could implement different node architectures. For example, one node could employ wavelength conversion but not TSIs while another node could implement TSIs but not wavelength converters.

WDM grooming networks can be classified into two categories [9]: dedicated-wavelength TDM (DW-TDM) networks and shared-wavelength TDM (SW-TDM) networks. In DW-TDM networks, each source-destination pair is connected by a *lightpath*, where a lightpath is defined as an all-optical connection between two nodes. Calls between the source and destination are multiplexed on the lightpath. If the bandwidth required by a new call at a node is not available on any of the existing lightpaths to the destination, a new lightpath to the destination is established. On the other hand, in SW-TDM networks, if a call cannot be accommodated on an existing lightpath to the destination, it is allowed to be multiplexed onto an existing lightpath to an intermediate node. The call is then switched from the intermediate node to the final destination either directly or through other nodes. However, if none of the existing lightpaths from the node can accommodate the call, a new lightpath to the destination is established. In this dissertation, we consider SW-TDM networks.

1.5 Issues and motivation

The issues in optical networks can be classified into three categories: technological, design, and operation. Technological issues involve research on components, such as couplers, amplifiers, switches, etc., that form the building blocks of the networks. With the help of these components, one then needs to design a network. Issues in network design include minimizing the total network cost, ability in the network to tolerate failures, scalability of the network to meet the future demands based on a projected traffic, etc. The operational part of the network involves monitoring the network for proper functionality, handling dynamic traffic in the network, reconfiguring the network in case of failure, etc. In this dissertation, we mainly concentrate on the network design and operation.

Network design involves assigning sufficient resources in the network that could meet the traffic demand. Typically, network design problems consider a static traffic matrix and aim at designing a network that would be optimized based on certain metrics. Network design problems employing static traffic matrix are typically solved as optimization problems. If the traffic pattern in the network is dynamic, specific traffic is not known a-priori, then the design problem involves assigning resources based on a certain projected traffic distributions. In the case of dynamic traffic the network design attempts to quantify certain performance metrics in the network based on the distribution of the traffic. The most commonly used metric in evaluating a network under dynamic traffic pattern is *blocking probability*. The blocking probability is computed as the ratio of number of requests that cannot be assigned a connection to the total number of requests. With this metric, one can make decisions on the amount of resources that need to be employed in a network, the operational policies such as routing algorithms, etc.

Simulations can be used to measure the blocking performance of networks, however require a long time. In order to expedite this process, analytical models are employed that serve as a coarse tool for evaluating network performance. The analytical models are employed in the network design phase to serve as an *elimination criteria* where a network design with lower performance projected by the analytical models are rejected. Therefore, analytical models form a critical component of the network design phase.

Analytical models for evaluating blocking probability in circuit-switched network have been in existence ever since the emergence of telephone networks. There are a wealth of resources available on the models of blocking probability developed for telephone networks, [10] and [11] for example. One key point that remains to be noted is that the blocking models developed for electronic circuit-switched networks assumed that the only reason for a request to be rejected is the lack of capacity on a link. The reason for this is because the transmission and the switching both were electronic, hence full-permutation switching from one link to another was possible.

With optical networks, the transmission of signals on the fiber has changed to the optical medium. In order to switch the connections in the optical domain, certain restrictions are imposed due to technological constraints. *Wavelength continuity constraint* that restricts a connection to occupy the same wavelength on every link of a chosen path from a source to destination is the key difference that led to the development of new analytical models for optical networks. It is to be noted that constraints such as wavelength continuity could result in rejecting a call even though the required capacity is available on all the links of the path. The reason for rejecting a request is due to the inability of the intermediate node to switch the connection between two links. Hence, newer models aim at modeling the switching capabilities at a node.

Analytical models for wavelength-routed networks were initially developed based on the assumption of statistical link load independence [12, 13, 14]. These models assumed that the traffic on a link is independent of other links in the network. Analytical models for quantifying the benefits of employing wavelength conversion capability in a network were later developed [15, 16, 17] with the assumption of statistical link load independence. Subramaniam and Somani [18] introduced the concept of link-load correlation and evaluated the blocking performance of networks with sparse-wavelength conversion, where only a few nodes in the network have full-wavelength conversion capability. Analytical models for multi-fiber networks were developed by extending the models for single-fiber wavelength-routed network [19, 20]. Models for limited-range wavelength conversion can be found in [21, 22, 23, 24, 25]. The analytical models assume fixed-path routing, i.e., the path that is chosen for establishing a connection from the source to destination is known apriori. Analytical models that account for dynamic routing based on up-to-date network status are complex, hence have received very little attention[26]. Analytical model for WDM/TDM networks can be found in [9, 27].

Almost all of the analytical models described above except [18] assume that all the nodes in the network employ similar switching architectures. In [18], a node can either implement either no wavelength conversion or full-wavelength conversion. Hence, the analytical models either have only one type of node architecture or at-most two. Analytical models for multi-fiber wavelength routed networks can be applied to WDM/TDM networks, however, the wavelength switches that are employed must have full-permutation switching capability. The major limitations of the analytical models developed thus far is that they are not generalized to consider multiple switching architectures at nodes on the path. Such a lack of generalization motivated the research presented in this dissertation.

1.6 Contributions of the dissertation

The goal of the dissertation is to provide a generalized network model that would allow us to model different switching architectures and develop a framework for connection establishment and blocking performance analysis on the generalized network model. The contributions of the dissertation, organized in various chapters, is as below:

In Chapter 2, we present our generalized network model called *Trunk Switched Network* (TSN). The generalized network model is based on grouping of channels on links, referred to as *trunks*, that could be switched at a node. Every node in the network views the links attached to them as trunks and is capable of switching channels among the same trunk. We illustrate the generalized nature of the proposed network model by examples where different optical node architectures are be modeled using the proposed network model.

In Chapter 3, we develop a framework for connection-establishment in a TSN. The framework is based on a matrix representation for links that stores the information about the channels on the link. We illustrate with examples the different kinds of information that could be obtained from a link and different methods of combining those information to obtain the information on a path. We complete the framework by presenting a methodology for path selection and channel assignment, equivalent of assigning resources on a link. We show that different path selection algorithms and wavelength/time slot/channel assignment algorithms developed earlier in the literature can be derived from this connection establishment framework.

In Chapter 4, we develop a framework for analysis of blocking performance in a TSN. We analyze the network blocking performance by considering the blocking performance of a path. We analyze a path by considering as it as two hops and applying iteration. Such an approach requires detailed modeling of only a two-link path, which is then modeled in a generalized manner to account for different switch architectures and multiple capacity requirement for traffic. We develop an analytical model for evaluating the blocking performance of establishing a multicast tree. We analyze the blocking performance for establishing a multicast tree by splitting it into multiple paths and combine the path blocking performance to obtain the blocking performance for establishing the tree. We develop a mapping methodology by which the distribution of trunks as viewed by one node can be translated to that viewed by another, enabling us to model a network with heterogeneous switching architectures. We also provide an iterative methodology for employing the analytical model to improve accuracy.

In Chapter 5, we validate the analytical model by comparing the results of blocking performance obtained using simulation. Through extensive simulation on different networks, we demonstrate the accuracy of the proposed analytical model. From the performance results, it is observed that smaller improvements in architectures could result in a significant improvement in performance. We show that the analytical model of a tree can be approximated with that of a path having the same number of links as in the tree. The analytical model assumes fixed-path routing strategy in the networks for reasons of tractability.

In Chapter 6, we consider dynamic routing strategies. We consider a wavelength-level grooming network in which the nodes can groom traffic only on the same wavelength. We propose a request-specific routing algorithm, called *Available Shortest Path* algorithm, that selects the shortest path among those that can accept the request. We propose a new mechanism to compute the effective network utilization in the network that would quantify the efficiency of a routing algorithm. We show that significant improvement in performance is achieved by employing routing mechanisms that consider request characteristics in selecting a path. We study the effect of *dispersity routing* that allows to split a connection into multiple smaller connections and assigning them channels on different wavelengths on the same path. We compare the effectiveness of this scheme versus increasing the grooming capability and show that dispersity routing could achieve significant performance without the need for increasing grooming capability. Dispersity routing is an inexpensive alternative as compared to increasing grooming capability at the nodes in the network.

In Chapter 7, we present our conclusions from the research conducted in this dissertation. We also list out the possibilities for future research and comment on applicability of the model in networks that do not necessarily employ the optical technology.

1.7 Summary

In this chapter, some background on optical networks was presented, specifically what motivated the deployment of today's WDM grooming networks. We identified important research issues and how various researchers have addressed these issues in the past. We identified that there is a need for a generalized network model that would enable modeling of optical network with heterogeneous grooming capabilities under a unified framework. The development of such a framework is the major contribution of this dissertation.

CHAPTER 2 Trunk Switched Networks

Networks of the past had both the medium of transmission and processing technology as electronics. Hence, the transmission and processing bandwidth at nodes were approximately of the same order. Electronic technology advanced simultaneously on the transmission and processing sides leading to a matched growth in the evolution of the networks. With the shift to the optical technology, the transmission capacity has taken a quantum leap while the processing capacity has seen only modest improvements in electronics. Optical processing is currently in the infancy stage and is not expected to be deployed widely in the near future. Hence, the backbone networks of the near future are expected to remain circuit-switched with a possibility of having optical switching at the intermediate nodes.

The increase in the transmission capacity in terms of multiple wavelengths each operating at few tens of gigabits per second with multiple time slots within a wavelength requires an equivalent increase in the electronic processing for efficient operation of networks. However, it is impractical to match the power of the optical technology with that of the electronic electronics if the nodes were to process the entire information that is received from different links it is connected to. Hence, the switching trends of the present-day and the near future revolves around having multiple simple processing devices that work independently on parts of the information that is received at a node. Such a network with nodes employing multiple independent processing elements that work independently on parts of the received information is a new paradigm that requires research. To this end, we make the first attempt in providing a generalized framework that would describe such a network. In this chapter, we propose a generalized network model called *Trunk Switched Network* (TSN). A TSN is a two-level network model in which the a link is considered as multiple channels and channels are combined together to form groups called *trunks*. We use this concept to develop a node architecture that is capable of grooming sub-wavelength level traffic over a link.

The chapter is organized as follows: Section 2.1 introduces the network architecture explaining the concept of trunks and channels. Section 2.2 describes the node-architecture in a TSN. The concept of free, busy, and available trunks are explained with example in Section 2.3. Section 2.4 provides examples highlighting the modeling of WDM grooming networks as TSNs.

2.1 Network architecture

A Trunk Switched Network (TSN) consists of nodes interconnected by links. Each link ℓ has a set of channels denoted by C_ℓ. A node views a link connected to it as groups of channels called *trunk*. A *trunk* at a node is defined as a unique non-null set of channels in a link. The number of trunks as viewed by a node i is denoted by K_i. The set of channels of a link ℓ that fall within a trunk x at node i is denoted by $\chi^i_{\ell,x}$. Let $S^i_{\ell,x} = |\chi^i_{\ell,x}|$ denote the number of channels in the set $\chi^i_{\ell,x}$. With the above definition of trunk, we have $\chi^i_{\ell,x} \cap \chi^i_{\ell,y} = \phi$.

2.2 Node architecture

Nodes in a TSN view the links as a set of trunks. Besides being a source or a destination of a connection in a network, a node can also act as an intermediate switching node for other connections that pass through it. Hence, the functionality of a node includes switching of channels from one link to another link in order to facilitate a connection. The switching architecture of a node in a TSN is shown in Figure. 2.1. The figure shows a node with four links. The trunks from the links are first isolated by trunk de-multiplexers. The isolated trunks are then fed into a stage of full-channel interchangers (FCI). The trunks from the different links are fed into their respective trunk switches. After the switching stage, the trunks are fed into a final stage of FCIs similar to that employed in the first stage. The trunks from the different switches are then combined using trunk multiplexers and sent out in the corresponding output links. One of the input and output links are assumed to be dedicated to the node to source and sink its own traffic.

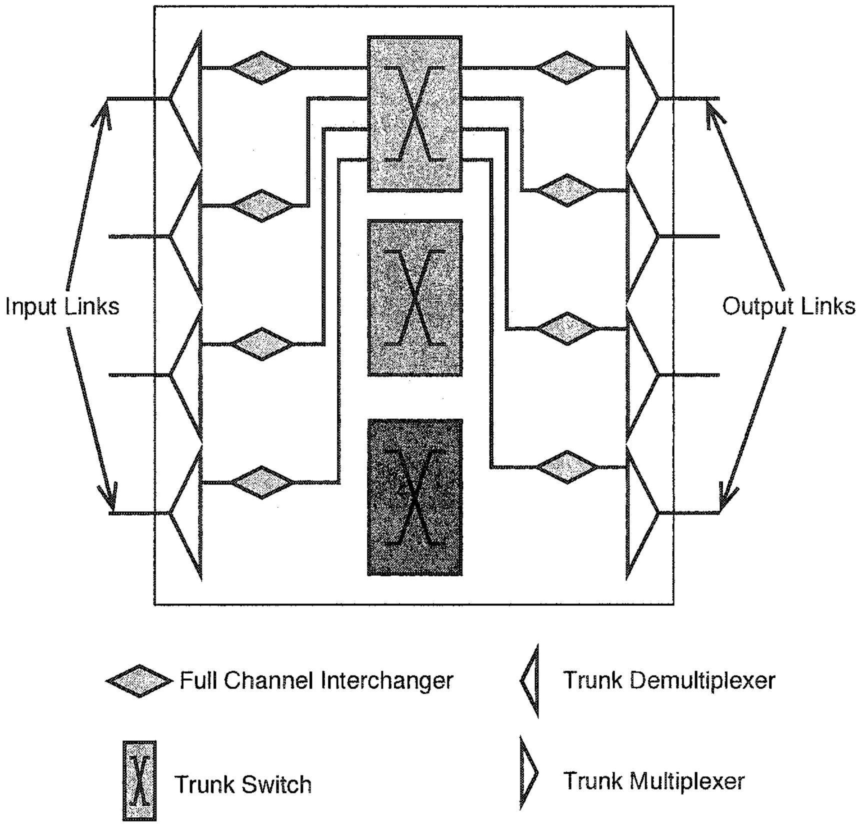


Figure 2.1 Node architecture in a trunk switched network.

The switching architecture employed at every node in a TSN obeys the following two conditions:

- A full-channel interchanger (FCI) for every trunk is employed at the input and output stages of the node, as shown in Figure. 2.1.
- Switching at a node obeys trunk-continuity constraint, i.e., the channels cannot be switched across trunks. Therefore, the trunk switches at a node function independently.

A full-channel interchanger (FCI) has the ability to convert any channel within the trunk on a link to any other channel within the same trunk on that link. Note that an FCI switches channels within the same trunk on a link and does not have the capability of switching the channels across links. This functionality allows a node to operate in an autonomous manner. For example, it is possible to re-arrange the channel assignments at a node without involving co-ordination with the neighboring node in order to optimize the network operation as long as the channel assignments on the links are kept the same. Another use of such a functionality is to employ specialized monitoring functions on a specific channel on the input link. The ability to re-arrange the channels inside the node allows the node to monitor different channels at different instants of time without co-ordinating with the neighboring nodes. The second constraint is employed due to the limited switching resources at a node. It is also assumed that every trunk switch is in a node has the same architecture.

The definition of a trunk could be different at different nodes. A TSN is said to be *homogeneous* if the collection of channels that constitute a trunk at a node is the same for all the nodes in the network. Otherwise, it is said to be *heterogeneous*. It is to be noted that the above definition does not specify any constraints on the switch architecture employed for each trunk at a node. The architecture of the trunk switches can be different at different nodes although all the nodes in the network view the links in the same manner.

2.3 Free and busy trunks

A channel on a link is said to be *busy* if it is allocated for a connection. Otherwise, it is said to be free. A trunk on a link at a node is said to be busy if all the channels on the trunk are busy. Otherwise, it is said to be free. The number of channels busy on a trunk at the input of a node is the same as the number of channels busy on the trunk at the input to the switch at the node. However, the distribution of the busy channels on the trunk at the input of the node may be different from that at the input of the switch. The number of trunks busy at the input of a node is the same as the number of trunks busy at the input of the switch at that node.

Now, consider a specific trunk at a node that acts as an intermediate switching node for a connection to be established. Figure. 2.2 shows the channel occupancy status of the trunk under consideration at the input and the output of the trunk switch at the intermediate node. The node has four incoming links. Note that the channel occupancy described in the figure denotes the channel status after the input FCI stage and before the output FCI stage.

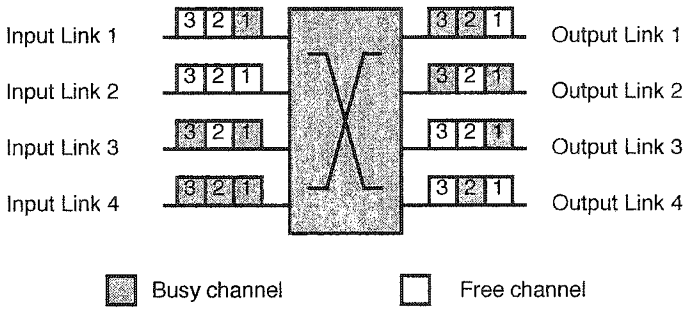


Figure 2.2 Channel occupancy at the input and output of a trunk switch at an intermediate node on a path. The channel occupancy shown here is after the FCI at the input stage and before the FCI at the output stage.

This trunk under consideration is said to be *available* on a two link path involving a certain link at the input of the switch and a certain link at the output of the switch if any free channel on the input link can be switched to any free channel on the specific output link. For example, if the trunk switch has full-permutation switching capability, then any free channel at any input link can be switched to any free channel at the output link but within the same trunk. In this case, if any of the first three links at the input and any of the output link is chosen, then the trunk is available on the two-link path. However, if link 4 at the input is chosen, then it has no free channels, hence the trunk is not available for the two link path. When a trunk switch has full-permutation switching capability, it has freedom to switch the channels in the space and channel domains.

If the trunk switch implements only a space switch, then a free channel on a link at the input must be switched to the same channel at any of the output link. Such switch does not have the flexibility to switch in the channel domain. Assume that link 1 at the input and output are the two links of the two-link path. At the input of the switch channels 2 and 3 are free while at the output they are not. As the channels can be switched only in the space domain (across links) but not in the channel domain, the trunk cannot be used for establishing a connection on the two link path involving the first link at the input and output. A trunk can be free at both the input links at a node, however it may not be available on a path having those two links as the trunk switch does not have the capability to switch the free channel at the input to a free channel at the output. If link 1 at the input and link 2 at the output are part of a path, then the channel 2 on the trunk is free, hence, the trunk is said to be available on the two-link path. A node employing space switches for every trunk is said to employ *channel-space switching (CS)* due to the FCI at the input stage of the node.

2.3.1 Sub-trunk assignment for multicast connections

Multicast connections are established in these networks by copying the signal in an input channel and switching the individual copies to multiple output channels. However, it is constrained that all the copies of the input signal should remain within the same trunk. Hence, this copying will also be referred to as *intra-trunk* copying. The total number of copies that can be made from an input signal is limited by the number of channels within a trunk. The number of copies that are made from an input signal is referred to as *degree of splitting*.

2.4 Modeling a WDM grooming network as a TSN

A single-fiber wavelength-routed WDM network employing W wavelengths can be modeled as W trunks with one channel per trunk. A multi-fiber multi-wavelength wavelength-routed network with F fibers and W wavelengths with no wavelength conversion can be viewed as W trunks with F channels per trunk. If full-wavelength conversion is available, then a link can be viewed as a single trunk with FW channels. However, networks that employ limited-wavelength conversion, as defined in [22] and [25], cannot be modeled easily or effectively as a TSN, as full-permutation wavelength-conversion is not employed.

Now, consider WDM grooming network. Let the links in the network employ three fiber, three wavelengths per fiber and two time slots per wavelength ($F = 3$, $W = 3$, $T = 2$). Figure 2.3 shows the eighteen channels that are available on a link. The shapes of the figures represent the time slots, the shades of the shapes represent wavelengths, and the number of shapes of a certain shade represents the number of fibers.

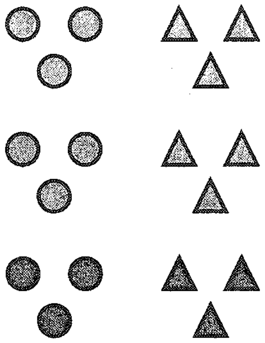


Figure 2.3 Representation of eighteen channels in a link having three fibers, three wavelengths per fiber, and two time slots per wavelength. Shapes represent time slots, shades represent wavelengths, while the number of shapes of a certain shade represents the fibers.

- If time slot interchange and wavelength conversion are not permitted, a node i views a link ℓ as WT trunks where each wavelength and time slot combination forms a trunk, i.e., $\chi^i_{\ell,(w,t)} = \{(l, f, w, t)|1 \leq f \leq F\}$, where $1 \leq w \leq W$ and $1 \leq t \leq T$. Every trunk has F channels as shown in Figure. 2.4(a).

- If time slot interchange is permitted, but not wavelength conversion, a node i views a link ℓ as W trunks where each wavelength forms a trunk, i.e., $\chi^i_{\ell,w} = \{(l, f, w, t)|1 \leq t \leq T \ and\, 1 \leq f \leq F\}$, where $1 \leq w \leq W$. Every trunk has FT channels as shown in Figure 2.4(b).

- If full-wavelength conversion is permitted, but not time slot interchange, then for a given link l, a time slot on all the wavelengths can be grouped to form a trunk, i.e., $\chi^i_{\ell,t} = \{(l, f, w, t)|1 \leq w \leq W \ and\ 1 \leq f \leq F\}$, where $1 \leq t \leq T$. Every trunk has has FW channels as shown in Figure 2.4(c).

- If both full-wavelength conversion and time slot interchange are permitted, then the entire link is treated as one trunk with FWT channels, as shown in Figure 2.4(d).

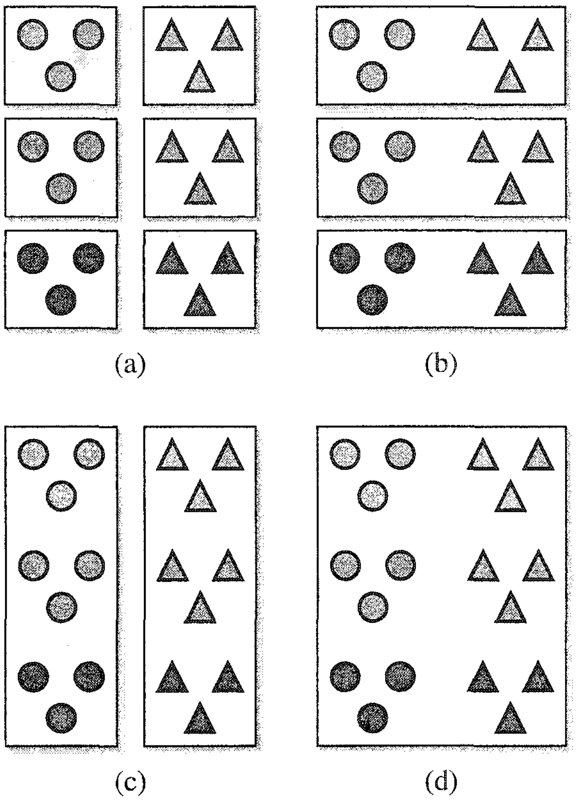


Figure 2.4 Possible grouping the channels in a link as trunks. (a) Wavelength-Time slot trunk (b) Wavelength trunk; (c) Time slot trunk; and (d) Link is a trunk.

2.5 Summary

In this chapter, we developed a network model called Trunk Switched Networks (TSN). Every link in a TSN is viewed as a set of channels. Nodes in a TSN combine the channels with similar properties into groups called *trunks*. We discussed the node architecture in a TSN and the classification of TSNs into homogeneous and heterogeneous networks based on the grooming architectures at nodes. We illustrated with examples the modeling of networks various grooming architectures as TSNs.

CHAPTER 3 Connection Establishment

Wide-area networks employing optical communication technology are expected to be circuit-switched in nature. The primary service provided by a circuit-switched network is to establish communication path between two nodes in the network. *Requests* or *calls* arrive at nodes in the network that require a connection of a certain bandwidth to another node. The setting up of a communication path between the nodes referred to as *connection establishment.*

In this chapter, we develop a framework for connection establishment in optical networks, called MICRON (Methodology for Information Collection and Routing in Optical Networks), emphasizing on methodology to maintain link information, combining link information to obtain path information, path selection, and sub-trunk and channel-assignment process. The chapter is organized as follows: Section 3.1 describes the steps involved in establishing a connection in circuit-switched networks and their classification. The prior work on connection-establishment in wavelength-routed WDM networks and WDM grooming networks are also discussed. An example network is described in Section 3.2 that will be used throughout this chapter for illustration. In Section 3.3 we develop the MICRON framework and illustrate with examples the features of the framework. Section 3.4 describes with an example the usage of the connection establishment framework to model a node employing channel-space switching.

3.1 Connection establishment in circuit-switched networks

Connection establishment in a circuit-switched network consists of two steps: *path selection* and *resource assignment.* Path selection refers to selecting a path from source to destination based on certain criteria. Resource assignment refers to assigning one or more channels depending on the requirement of the request on every link of the chosen path.

3.1.1 Path selection

The first step of the connection establishment process, namely path selection, can be carried out in several ways. If a source-destination pair has one pre-selected path, then it is referred to as *fixed-path* approach. If a path is selected depending on the network status from a pre-selected set of candidate paths, then it is referred to as *alternate path selection*. The set of candidate paths remain the same at all times and do not change with the network status. If the candidate paths are chosen based on the network status, the path selection process is referred to as *dynamic routing*. Dynamic path selection and dynamic routing approaches require network state information to be maintained at different nodes in the network.

Up-to-date network state information is collected in the network either though *link-state* or *distance-vector* protocols. In link-state protocol, every node in the network transmits to every other node its node information and the information of the links that it is connected to. Hence, every node in the network has the precise knowledge on the topology and the current status of the network. The main drawback of this approach is that it is not scalable. As the network size increases, the amount of information that needs to be maintained at a node also increase. Hence, it is impractical to adopt this approach for large networks. Wide-area optical networks are expected to employ only a few nodes. Therefore in the context of optical networks, employing link-state protocols is still a preferred method of information collection.

Distance-vector protocols maintain up-to-date network information by exchanging the node and link information with neighbors. Nodes in a network employing such an approach for information collection do not have the knowledge of the network topology. Every node maintains a routing table that would indicate one or more preferred neighbor to reach a destination node. The main drawback of this approach is the scalability, however due to a different reason as compared to the link-state approach. As the changes in the network information is propagated through a series of neighborhood information changes, it takes a significant amount of time for the changes in one part of the network to be reflected in the other regions. The time required for a change in the network to be propagated increases with increase in the network size. Also, such an approach does not always necessarily result in the convergence of the network state as viewed by the nodes.

A preferred approach to path selection in large networks is to employ alternate path selection. Selecting a path from a fixed set of candidate paths requires information to be collected on those candidate pairs. This information can be collected as a part of the connection establishment procedure by sending requests along all the candidate pairs. Note that such an approach can be employed after a request arrival at a node, hence the information that is collected can be tailored to the requirements of the call.

Path selection algorithms are also classified based on how the path selection decisions are made. If the source node selects a path to the destination, then it is referred to as *source routing*. Note that such a routing in the network requires the source to have entire knowledge about the network to make the decision. Hence, networks that employ source routing also employ link-state protocol for information collection. If the path selection is done independently at different nodes, it is referred to as *distributed routing*. In networks that employ distance-vector protocols for information collection, nodes do not have the knowledge of the entire network topology. Hence, requests for a connection to a certain node are forwarded to a preferred neighbor node. The neighboring node then makes the decision on the next hop of the connection. In such an approach, the source node does not have control over the connection that is established in the network. Both the above mentioned approaches have their own advantages and dis-advantages. In source routing, the source node has complete control over the established path. Such a control is an advantage when the node attempts to include or avoid certain nodes in the path or implement quality of service requirements. However, the major drawback of source routing is the requirement on the knowledge of the entire network to select a path. Distributed routing has its advantage that an intermediate route could reroute the connection request depending on the current network status, thus adapting to the network changes more easily as compared to source routing. However, the disadvantage is that the source node does not have the control over the path that is chosen.

Path selection can be made either before or after a request arrival. Algorithms that select a path before a request arrival typically select the path such that most of the requests can be accommodated. Such an approach to path selection is referred to as *destination-specific*, where a path is selected based only on the destination node. Algorithms that select paths after the arrival of a request can prune the set of candidate paths based on the request characteristics. Such algorithms are referred to as *request-specific* algorithms. The paths that are selected from a specific source to destination could be different for requests with different requirements. The former approach requires continuous monitoring of the network status. Although a path is readily available when a request arrives when destination-specific approach is employed, a connection may still not be established as changes in the network need not be reflected immediately. If the network has slow-varying dynamics, then such situations are rare. Hence, a destination-specific approach would have a lower connection establishment time as compared to a request-specific approach.

3.1.2 Resource assignment

The second step of connection establishment process, namely resource assignment, can have many versions as well. On a chosen path, one or more channels can be assigned on every link that is either based on some global metric or just based on the available resources on the path. For example, in a wavelength-routed network, resource assignment refers to wavelength assignment. Wavelength assignment algorithms such as random wavelength assignment or first-fit wavelength assignment assigns resources based on the status of path. Wavelength assignment algorithms such as most-used or least-used wavelength assignment policies assign a wavelength based on the wavelength usage across the entire network.

3.1.3 Prior work in connection establishment in WDM grooming networks

The two steps in connection establishment has been referred to as *routing and wavelength assignment* (RWA) in the context of wavelength-routed networks and has received extensive attention from the research community in the early years of optical networking research. Several RWA algorithms have been proposed for routing static traffic demands to optimize the network capacity by formulating them as integer linear programming (ILP) problems. While such an approach could produce the optimal solution for static traffic demands, applying these techniques to dynamic traffic is not practical due to their prohibitively large computation time. Such an approach would require much more computational complexity when sub-wavelength traffic is considered. Hence, heuristic based solutions are preferred for connection establishment under dynamic traffic scenario.

Path selection in wavelength-routed WDM networks has been studied extensively in the literature. Wavelength assignment in networks employing fixed-path routing has been analyzed in [28]. Routing in networks with more than one candidate path has been shown to offer significant performance improvement[29]. Fixed alternate path routing (FAPR) has been studied in [30] and [31]. Fixed-path least-congestion routing (FPLCR) has been analyzed in [32]. In these approaches, a path from a source to destination is selected from a set of pre-computed paths. While FAPR attempts the paths in a specified order, FPLCR selects the least loaded path. In [33], a wavelength-routed network with W wavelengths and no wavelength conversion is treated as W networks with one wavelength each. Shortest path algorithm is applied on each network. A request from a source to destination is assigned a connection on a wavelength that has the minimum path length. The impact of distributed routing on the connection setup time and stabilizing time for exchanging of network state has also been analyzed in [33]. In [34], alternate link routing (ALR) is proposed. In this approach, a pre-computed set of preferred links to reach a destination is available at every node. A request is forwarded on any one of the preferred outgoing links to the destination. In [26], an analytical model is developed for evaluating the blocking performance of various routing algorithms, including adaptive unconstrained routing which does not restrict the path selection to any pre-defined set of routes. Mechanisms for controlling setup and tear-down of lightpaths and network update procedures for wavelength-routed networks are presented in [35]. A survey on routing and wavelength assignment algorithms in wavelength-routed WDM networks can be found in [36].

Routing, wavelength and time-slot assignment, referred to as RWTA, in WDM grooming networks has received very little attention in the research community [37]. Earlier work on WDM grooming networks concentrate on performance analysis of fixed-path (shortest-path) routing strategies with random wavelength assignment [27]. However, there has been very little or no significant research can be found in the literature on dynamic routing in WDM grooming networks.

3.2 Example network

Consider the example two paths from node 1 to 5 in a network shown in Figure 3.1. Let the nodes be connected using links employing three fibers each carrying three wavelengths and two time slots per wavelength. Also assume that nodes 1, 3, 6, and 7 are wavelength-level grooming nodes; nodes 2 and 5 are time-slot-level grooming nodes; and node 4 is a full-grooming node. Wavelength-level grooming nodes view the link as 3 wavelength trunks (denoted by W_1, W_2, and W_3) with 6 channels in each, time slot-level grooming nodes view a link as two time slot trunks (denoted by T_1 and T_2) with 9 channels in each, and a full-grooming node views a link as one trunk (denoted by F_1) with 18 channels.

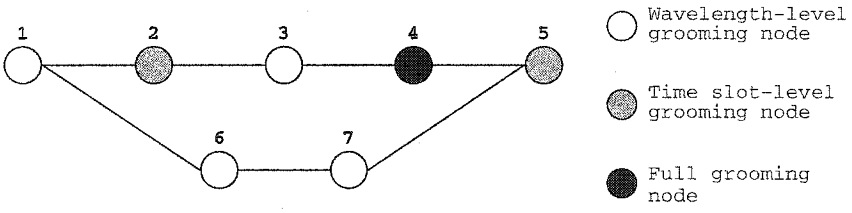

1
2
3
4
5
6
7
○ Wavelength-level grooming node
◍ Time slot-level grooming node
● Full grooming node

Figure 3.1 An example network showing two paths from node 1 to node 5.

Figure 3.2 shows the expanded view of the network indicating the different trunks at the nodes. For example, consider trunk W1 of node 1 and trunk T1 of node 2. Let ℓ_{12} denote the link that connects node 1 to 2. The number of channels in the link ℓ_{12} that belongs to both the trunk W_1 at node 1 and T_1 at node 2 is 3. The corresponding three channels are $(\ell_{12}, 1, 1, 1)$, $(\ell_{12}, 2, 1, 1)$ and $(\ell_{12}, 3, 1, 1)$, each channel belonging to a distinct fiber. The arrow connecting trunk W1 of node 1 to trunk T1 of node 2 indicates the number of free channels that belong to both the trunk definitions.

A value of 3 indicates that all the channels belonging to both the trunk definitions are free. An arrow connecting a trunk at a node to a trunk at its neighboring node refers to a sub-trunk. Note that, if there are no channels present in a sub-trunk, then the arrows are not shown. For example, consider nodes 6 and 7. Both the nodes are wavelength-level grooming nodes. Therefore, there are no channels that would belong to a sub-trunk Θ_{xy}^{67}, where $x, y \in \{W_1, W_2, W_3\}$ and $x \neq y$. Hence, the corresponding arrows are not shown in the figure.

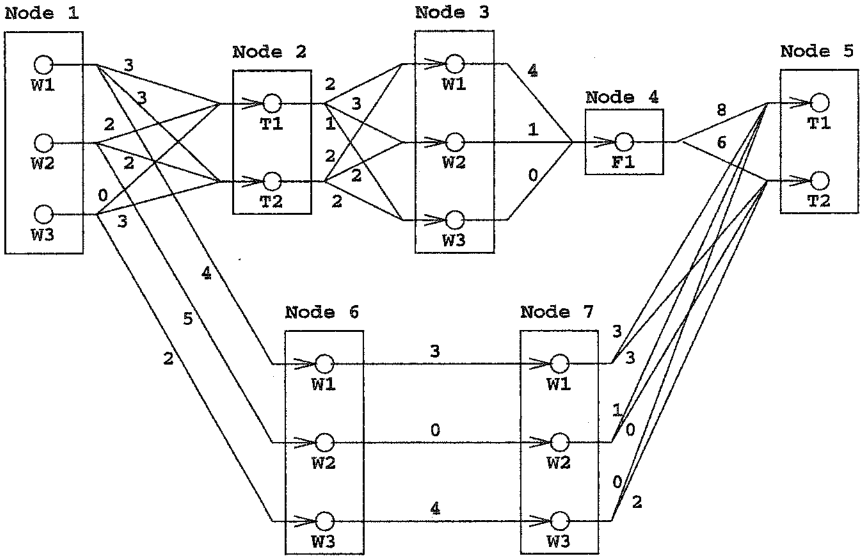


Figure 3.2 Expanded view of the network with channel occupancy information.

Assume that the network is observed at some instant of time during its operation and the channel occupancy in the links are known. Let $\psi(l, f, w, t)$ denote the status of the channel: denoted by 0 if occupied by a connection, 1 if the channel is free. Let $(l, f, w, t).Status$ denote if the channel is assigned for a primary (or working) connection (denoted by PRIMARY) or a backup connection (denoted by BACKUP). Let $(l, f, w, t).BackupList$ denote the set of requests that share the channel (l, f, w, t) for their backup paths. This implies that the primary paths of these requests are link-disjoint.

Consider a request, say r, with a primary path already assigned. Let $(l, f, w, t).LinkDisjoint(r)$ indicate if the request r has its primary path link-disjoint with the primary paths of all the requests that have their backup assigned on the channel (l, f, w, t) (denoted by 1 if it is link disjoint, 0 otherwise). If the channel is not assigned to either a primary or a backup connection, the value is set to 0. Such an information strictly identifies if the channel can be "shared" with already allocated backup connections.

3.3 MICRON Framework

We develop a framework for connection establishment in TSN. The framework includes the representation of link information, combining them to obtain the path information, selection of a path, and sub-trunk assignment. The steps are described in detail in the following subsections.

The notations employed in developing the framework for the framework are listed in Table 3.1.

Table 3.1 Notations employed in the MICRON framework.

Notations	Description
K_i	Number of trunks as viewed by node i.
S_i	Number of channels per trunk at node i.
$\chi^i_{\ell,x}$	Set of channels on link ℓ that fall within trunk x at node i.
$\Theta^{ij}_{x,y}$	Set of channels on link ℓ connecting node i to j that fall within trunk x at node i and trunk y at node j. (= $\chi^i_{\ell,x} \cap \chi^j_{\ell,y}$)
L_{ij}	Information matrix for link $i - j$ (Dimension = $K_i \times K_j$)
P_{sd}	Path information matrix for a specific path from node s to node d having one or more links. (Dimension = $K_i \times K_j$)
U_i	Unit row vector at node i (Dimension = $1 \times K_i$). All entries are 1.
I_i	Identity matrix at node i (Dimension = $K_i \times K_i$).
V_{sd}	Path information vector for a specific path from node s to node d having one or more inks (Dimension = $1 \times K_d$). $V_{sd} = U_s P_{sd}$

3.3.1 Link information

A link in a TSN connects two nodes that could view the channels in a link as different groups of trunks. The information about the channels representing each sub-trunk on a link is organized as a matrix. A link connecting node i and j is represented by a matrix L_{ij} as:

$$L_{ij} = \begin{bmatrix} l_{11} & l_{12} & \dots & l_{1K_j} \\ l_{21} & l_{22} & \dots & l_{2K_j} \\ . & & & \\ . & & & \\ l_{K_i1} & l_{K_i2} & \dots & l_{K_iK_j} \end{bmatrix} \tag{3.1}$$

where each element l_{xy} denotes a certain property about the channels in the link that belong to the sub-trunk Θ_{xy}^{ij}. For example, consider the Link 1–2 in the example network shown in Figure 3.1. Node 1 views each wavelength as a trunk, hence has 3 trunks. node 2 views each time slot as a trunk, hence has 2 trunks. Hence, L_{12} is a 2 × 3-matrix.

The matrix can denote different properties of the channels that belong to a certain sub-trunk. We discuss two specific examples in this chapter to illustrate the effectiveness of the proposed framework.

Case 1: Connectivity

Assume that a connection request that arrives at a node requires a bandwidth of B channels. In order to determine if a link has enough capacity in each of its sub-trunk, every element l_{xy} of the matrix L_{ij} is denoted by 1 if the number of free channels that belong to a sub-trunk Θ_{xy}^{ij} has a capacity of at least B. The matrix L_{ij} is defined as:

$$l_{xy} = \begin{cases} 1 & \text{if } \left(\sum_{(l,f,w,t)\in\Theta_{xy}^{ij}} (l,f,w,t).Availability\right) \geq B \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where $1 \leq x \leq K_i$ and $1 \leq y \leq K_j$. It is observed that the matrix gives the connectivity information to route a call that requires a capacity of B without splitting the connection across sub-trunks. For the example network considered in Figure 3.1, the link information matrices for different links for connection requests of one channel capacity is shown in Figure 3.3.

Case 2: Available Capacity

In order to represent the available capacity on each sub-trunk of a link, every element l_{xy} of the

$$L_{12} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \quad L_{23} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad L_{34} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$L_{45} = \begin{bmatrix} 1 & 1 \end{bmatrix} \, L_{16} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad L_{67} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$L_{75} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Figure 3.3 Link information matrices indicating if there is at least one free channel in a sub-trunk.

matrix L_{ij} is defined as the number of free channels that belong to a sub-trunk Θ_{xy}^{ij} as:

$$l_{xy} = \sum_{(l,f,w,t)\in\Theta_{xy}^{ij}} (l, f, w, t).Availability \tag{3.3}$$

where $1 \leq x \leq K_i$ and $1 \leq y \leq K_j$. The matrix representation for different links in the example network in Figure 3.2 are shown in Figure 3.4.

$$L_{12} = \begin{bmatrix} 3 & 3 \\ 2 & 2 \\ 0 & 3 \end{bmatrix} \quad L_{23} = \begin{bmatrix} 2 & 3 & 1 \\ 2 & 2 & 2 \end{bmatrix} \quad L_{34} = \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix}$$

$$L_{45} = \begin{bmatrix} 8 & 6 \end{bmatrix} L_{16} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad L_{67} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$L_{75} = \begin{bmatrix} 3 & 3 \\ 1 & 0 \\ 0 & 2 \end{bmatrix}$$

Figure 3.4 Link information matrices indicating the number of free channels in a sub-trunk.

Note that the matrices obtained using the available sub-trunk capacity also contains the information of the matrices representing connectivity information. Depending on the level of information that is required in the network, different matrix representations can be employed.

Case 3: Backup sharing

In this case, it is assumed that a path for establishing a primary connection has been selected. A backup path for the request has to be computed, hence the matrix is filled with information related to the backup capacity that is freely available. This implies that the capacity has already been allocated as backup to some other connection in the network and the current request can be overloaded on that capacity as their primary paths are link disjoint. The matrix L_{ij} is defined as:

$$l_{xy} = \sum_{(l,f,w,t)\in\Theta^{ij}_{xy}} (l, f, w, t).LinkDisjoint(r) \tag{3.4}$$

For example, assume that the link information depicted in Fig. 3.2 as the capacity that has already been allocated as backup to other connections in the network with which the backup for the request under consideration could be shared. The matrix representation in this case would be the same as that described for case of available information in Fig. 3.4.

3.3.2 Path information

The information about a certain path from a node i to k that are not physically connected by a fiber is obtained by combining the link information in the path. The matrix representation for a path is defined in a manner similar to that of a link. A path matrix from node i to k through j is obtained as a matrix multiplication of individual path segments P_{ij} and P_{jk} as:

$$P_{ik} = P_{ij}P_{jk} \tag{3.5}$$

We employ a generalized version of matrix multiplication to compute the path metric. An element p^{ik}_{xy} (the superscript ik denotes the matrix to which the element belongs to) is obtained as:

$$p^{ik}_{xy} = (p^{ij}_{x1} \otimes p^{jk}_{1y}) \oplus (p^{ij}_{x2} \otimes p^{jk}_{2y}) \oplus ... \oplus (p^{ij}_{xK_j} \otimes p^{jk}_{K_jy}) \tag{3.6}$$

The operators $\otimes$ and $\oplus$, denoted as a tuple $(\otimes, \oplus)$, can be defined in different combinations so that several meaningful results are obtained. It can be observed that when $\otimes$ is integer or real number multiplication and $\oplus$ is integer or real number addition operation, the above equation denotes the traditional matrix multiplication.

To illustrate the significance of different operators, we consider the two example matrix representation of links, namely the connectivity and available capacity matrix representations, and apply two different set of operators to obtain different information from the network.

Case 1: Arithmetic operators

In this case, we consider the integer multiplication ($\times$) and integer addition ($+$) as the operators for $\otimes$ and $\oplus$, respectively. Consider the matrix representation shown in Figure 3.3 for a request that requires connection of one channel capacity. Applying the operator on the path 1–2–3–4–5, we obtain the path information matrix as:

$$P_{1-2-3-4-5} = \begin{bmatrix} 4 & 4 \\ 4 & 4 \\ 2 & 2 \end{bmatrix} \tag{3.7}$$

An element p_{xy} of the above matrix denotes the number of distinct sub-trunk selections available from trunk x of node 1 to trunk y of node 5. For example, there are four paths that can start at trunk W1 of node 1 and end at trunk T1 of node 5. These four trunk assignments on the path are represented as a set of tuples containing node numbers and the trunk number on that node through which the connection passes through. The four possible trunk assignments on the path, denoted by P_1 through P_4 are:

$$\begin{aligned} P_1 &: \ \{(1, W_1), (2, T_1), (3, W_1), (4, F_1), (5, T_1)\} \\ P_2 &: \ \{(1, W_1), (2, T_1), (3, W_2), (4, F_1), (5, T_1)\} \\ P_3 &: \ \{(1, W_1), (2, T_2), (3, W_1), (4, F_1), (5, T_1)\} \\ P_4 &: \ \{(1, W_1), (2, T_2), (3, W_2), (4, F_1), (5, T_1)\} \end{aligned}$$

The existence of trunk assignment for other trunk pairs can be easily verified from Figure. 3.1. Consider the link information as shown in Figure 3.4. Applying the operator $(\times, +)$ on these matrices results in the path information matrix for path 1–2–3–4–5 as:

$$P_{1-2-3-4-5} = \begin{bmatrix} 504 & 378 \\ 336 & 252 \\ 240 & 180 \end{bmatrix} \tag{3.8}$$

An element p_{xy} of the above matrix denotes the number of possible channel assignment combinations on the path that start at a certain trunk x on node 1 and end at a trunk y on node 5. For example, consider the possible trunk assignments that start at trunk W_1 at node 1 and end at trunk T_1 at node 5. On every sub-trunk on the path the number of ways of assigning a channel is the same as the number of channels in the sub-trunk. Hence, the number possible channel assignments on a specific trunk assignment on the path is the product of the number of channels on the assigned sub-trunk on every link. The number of possible channel assignments on the four possible trunk assignments P_1 through P_4 that start the connection at trunk W_1 at node 1 and end at trunk T_1 at node 2 are 192, 72, 192, and 48, respectively, adding up to 504 possible ways of channel assignment.

Case 2: Selection operators

In this case, we assume that the operator $\otimes$ indicates the minimum of the two operands while the operator $\oplus$ indicates the maximum of the two operands. Applying this set of operation to the matrix representation in Figure 3.3 for connectivity, we obtain the matrix representation for the path 1–2–3–4–5 as:

$$P_{1-2-3-4-5} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{3.9}$$

The elements of the matrix indicate the existence of a channel allocation scheme for one channel capacity call that would start at trunk x at node 1 and end at trunk y at node 5. Note that matrix in Equation. (3.9) can be obtained from the matrix in Equation. (3.7) or (3.8) by replacing every element in the matrix by 1 if it is non-zero.

Applying this set of operation to the matrix representation in Figure 3.4, we obtain the maximum capacity that can be routed from node 1 to node 5 without splitting the connection. The matrix representation for the path is obtained as:

$$P_{1-2-3-4-5} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 2 & 2 \end{bmatrix} \tag{3.10}$$

Consider the possible trunk assignments that start the connection at trunk W_1 at node 1 and end at trunk T_1 at node 2. Consider the four possible trunk assignments P_1 through P_4. It is observed from Figure 3.1 that the trunk assignments P_1 and P_3 have the link connecting node 2 to 3 as bottleneck with two channel capacity. The trunk assignments P_2 and P_4 have the link connecting node 3 to 4 as bottleneck with one channel capacity. Hence, a maximum of two-channel capacity connection can be routed from node 1 to 5 starting at trunk W_1 at node 1 and ending at trunk T_1 at node 5.

In the link information depicted in Fig. 3.2 is assumed to indicate the capacity available on each sub-trunk that has already been assigned to other connections as backup and could be shared with the request that is under consideration, employing the operator set (min, max) would in a path matrix as shown in Equation 3.10. In this scenario, the elements of the matrix denote the maximum capacity that could be shared on the path by the request.

3.3.3 Two-pass approach to connection establishment

When a call arrives at a node, a request for connection establishment is sent along a set of candidate paths. The connection establishment is carried out in two passes: *Forward pass and Reverse pass* [38]. During the forward pass, the connection request is forwarded to the nodes along the path along with a vector, called Path Information Vector (PIV). The path information vector at a node k for a path p with source i and destination k, denoted by V_{ik} is of dimension $1 \times K_k$. V_{ik} is obtained as a product of the path information vector at the source node and the information matrix of the path connecting nodes i and k:

$$V_{ik} = U_i P_{ik} \tag{3.11}$$

where U_i denotes the path information matrix at the source node which is always set as a unit row vector.

Assume that in the path from node i to k passes through node j. Re-writing the above equation gives the relationship between the PIV vectors at node j and node k.

$$\begin{aligned} V_{ik} &= U_i P_{ik} && (3.12) \\ &= U_i P_{ij} P_{jk} && (3.13) \\ &= V_{ij} P_{jk} && (3.14) \end{aligned}$$

The matrix-vector multiplication employed above is similar to the generalized matrix multiplication proposed earlier in the paper with the operator tuple $(\otimes, \oplus)$. The elements of PIV at a node indicates specific properties about paths that end at a certain trunk. For example, if the link information matrix represented in Figure 3.3 and operator $(\times, +)$ are employed, then the resulting PIV at each node indicates the number of possible trunk assignments on the path that would terminate the connection on a certain trunk at that node.

During the forward pass of the connection establishment, a node j on the path p with source i can forward either the path information matrix L_{ij} to its neighboring node or the path information vector P_{ij}. Forwarding the latter has the advantage of minimizing the amount of information forwarded. Note that the reduction information exchange will be significant when the number of trunks at a node is large. Hence, the path information vector at a node is known to the successive node in the path and will be used to assign a sub-trunk on the reverse pass.

3.3.4 Path selection

The path information vector can be used to select a suitable path from a given source-destination pair.

For example, consider the two paths from node 1 to 5: 1–2–3–4–5 and 1–6–7–5. Employing the matrix information represented in Figure 3.3 and operator $(\times, +)$, we obtain the path information vector for the two paths as:

$$\begin{aligned} V_{1-2-3-4-5} &= \begin{bmatrix} 10 & 10 \end{bmatrix} \\ V_{1-6-7-5} &= \begin{bmatrix} 1 & 2 \end{bmatrix} \end{aligned}$$

With these matrices known at the destination, one could employ different comparison algorithms to select a path. For example, the total number of trunk assignments possible on a path is obtained by summing all the elements of the matrix. A path that has the maximum value for this metric can be chosen for establishing the connection in order to distribute the traffic in the network.

If the matrix representation in Figure 3.4 and operator (min, max) are employed, the path information vector for the two paths are obtained as:

$$\begin{aligned} V_{1-2-3-4-5} &= \begin{bmatrix} 2 & 2 \end{bmatrix} \\ V_{1-6-7-5} &= \begin{bmatrix} 3 & 3 \end{bmatrix} \end{aligned}$$

It can be observed that the path 1–6–7–5 can route a call for three channel capacity request without splitting, while the other path cannot. Hence, if traffic requirements in the network are diverse and destination-based path-selection is employed, then the path 1–6–7–5 could be chosen so as to minimize the blocking at that instant of time.

The selection of the path need not be based only on the path information vector. Different metrics such as hop-length, delay, abstract cost, etc. that could be included for link state vector and possible path selection schemes for WDM grooming networks are discussed in [39]. The path information vector can also be extended to include multiple metrics in order to select a path.

3.3.5 Sub-trunk assignment

At the end of the forward pass, the destination node has the path information vector for the different probed paths and selects a path based on a certain path selection algorithm. Once a path is chosen, a sub-trunk has to be selected on every link of the path in order to complete the connection establishment. The sub-trunk assignment is carried out in two steps as: (1) The destination node first selects the trunk at its node where the connection would terminate; and (2) Every node in the network selects the output trunk at its previous node. If a link connects node i and j, then the node j selects the output trunk at node i, hence the sub-trunk assignment on the link i–j.

Consider the information matrix represented in Figure 3.3 and operator $(\times, +)$. The path information vectors obtained at different nodes are shown in Figure 3.5.

$$
\begin{aligned}
V_{11} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \\
V_{12} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \end{bmatrix} \\
V_{13} &= \begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 5 & 5 \end{bmatrix} \\
V_{14} &= \begin{bmatrix} 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \end{bmatrix} \\
V_{15} &= \begin{bmatrix} 10 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 10 \end{bmatrix}
\end{aligned}
$$

Figure 3.5 Path information vector computed at the nodes along the path 1–2–3–4–5.

The trunk assignment to end the connection at the destination node can be made using the path information vector. Several trunk selection schemes such as first-fit, best-fit, random, etc. could be employed. In this paper, we illustrate the random sub-trunk assignment. Let x_k denote the trunk that is chosen to accommodate the connection at node k.

In order to select a sub-trunk on link $j - k$, a *ratio vector* is computed at node k. The vector, denoted by R_{jk}, is obtained as the product of the vector at the path information vector previous node, V_{ij} and the column vector of the link information matrix L_{jk} corresponding x_k:

$$
\begin{aligned}
R_{jk} &= V_{ij} \times L_{jk}^T(x_k) && (3.15) \\
&= \begin{bmatrix} v_1 & \dots & v_{K_j} \end{bmatrix} \circ \begin{bmatrix} l_{1x_k} & \dots & l_{K_j x_k} \end{bmatrix} && (3.16)
\end{aligned}
$$

$$= \begin{bmatrix} v_1 \circ l_{1x_k} & \dots & v_{K_j} \circ l_{K_j x_k} \end{bmatrix} \tag{3.17}$$

where $L^T_{jk}(x_k)$ denotes the transpose of the column vector corresponding to the column x_k of the matrix L_{jk} and the operator $\circ$ denotes the element-wise operation on the row vectors. Again, one could define different operators depending on the construction of the information matrix. Since the input trunk at node k is decided, the choices of output trunk at node j is also dictated by the channel occupancy of the channels that fall within $\Theta^{jk}_{yx_k}$. The output channel at node j can be selected in various ways using the ratio vector.

During the reverse pass, a sub-trunk is allocated on the path. As node 5 is the destination, it selects a trunk for the connection to terminate. We illustrate the random sub-trunk assignment here. We assume that the operator $\circ$ denotes integer multiplication[1].

The PIV at node 5 indicates that there are 20 possible sub-trunk assignments with each trunk being able to terminate 10 each. Hence, one of the two is chosen with equal probability. In general, if p_x sub-trunk assignments are possible on the path that would terminate the connection at the destination node at trunk x, then the trunk x is chosen with a probability $\frac{p_x}{\sum_{y=1}^{K_d} p_y}$, where K_d denotes the number of trunks at the destination node d. In the example considered here, one of the two trunks is selected with equal probability. Assume that the trunk chosen is $T2$. The node also selects the output trunk at its previous node. In order to select this, the ratio vector is computed as:

$$R_{45} = \begin{bmatrix} 10 \end{bmatrix} \circ \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 10 \end{bmatrix} \tag{3.18}$$

In this case, as only one output trunk is available, it is selected. Hence, on link 4–5, a channel that belongs to trunk $F1$ of node 4 and trunk $T2$ of node 5 is selected. node 5 confirms the selection of output trunk $F1$ to node 4 during the reverse pass in the network.

As the trunk assignment for the connection at node 4 is decided by node 5, node 4 chooses the output trunk at node 3 by computing a similar ratio vector as:

$$R_{34} = \begin{bmatrix} 5 & 5 & 5 \end{bmatrix} \circ \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 5 & 0 \end{bmatrix} \tag{3.19}$$

[1] The operator $\circ$ is the same as operator $\otimes$ since the element-wise operation that is evaluated here is similar to the matrix-vector multiplication employed for computing path information vector.

R_{34} vector denotes the selection ratio for the three output trunks at node 3. Note that although there are 5 possible paths that could end at trunk $W3$ at node 3, there are no free channels on link 3–4 that fall within trunk W_3 of node 3 and trunk F_1 of node 4. This information is reflected in the selection ratio vector SR_{34} as a zero entry corresponding to the ratio for trunk W3. Hence, trunk W1 or W2 is selected with equal probability. Assume that trunk W2 is selected in this case.

At node 3, the vector R_{23} is computed as:

$$R_{23} = \begin{bmatrix} 2 & 3 \end{bmatrix} \circ \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \end{bmatrix} \tag{3.20}$$

node 3 selects the output trunk at node 2 in the ratio of 2:3, i.e, trunk $T1$ is selected with a probability of 0.4 while trunk $T2$ is selected with a probability of 0.6. Assume that trunk $T1$ is chosen.

At node 2, the vector R_{12} is computed as:

$$S_{12} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \tag{3.21}$$

One of the trunks W_1 or W_2 is chosen with equal probability. Assume that W_1 is chosen. This selection is sent to node 1 completing the sub-trunk assignment. Now, the path established for the connection can be written as a set of node-trunk pair assigned at each node on the path $\{(1, W_1), (2, T_1), (3, W_2), (4, F_1), (5, T_2)\}$ or equivalently as a set of link and sub-trunks pair on the path $\{(\ell_{12}, \Theta_{W_1T_1}), (\ell_{23}, \Theta_{T_1W_2}), (\ell_{34}, \Theta_{W_2F_1}), (\ell_{45}, \Theta_{F_1T_2})\}$. Any channel belonging to the sub-trunk assigned at a link can be chosen for establishing the channel as every node has full-permutation switching capability within a trunk.

It can be observed that such a trunk selection strategy selects with uniform probability a possible sub-trunk assignment on the path. In the above trunk selection process, if the information matrix representation shown in Figure 3.4, operator $(\times, +)$, operator $\circ$ set to multiplication, and random channel assignment within a chosen trunk on each link, then the resulting channel assignment algorithm selects a channel uniformly from the set of all possible channel assignments possible on the path.

Several other channel trunk selection approaches can be developed based on the proposed connection establishment framework. Selecting the trunk that has the minimum or maximum value in the selection ratio vector would have effect similar to packing and spreading the connections across the available sub-trunk assignments in the path. In order to implement a first-fit sub-trunk assignment the first available trunk is chosen from the ratio vector. It can be easily observed that the routing algorithms that have been proposed earlier in the literature for wavelength-routed networks can be easily derived from the proposed framework.

3.4 Modeling a channel-space switch using the connection establishment framework

The connection establishment framework proposed in this chapter can be employed to simulate a network where the nodes employ channel-space switches. Modeling such a switching architecture is achieved by incorporating dummy nodes in the network.

Figure. 3.6(a) shows a node in a network with an incoming and out-going link. Assume that this node implements channel-space switching. This node is modeled with two dummy nodes as shown in Figure. 3.6(b).

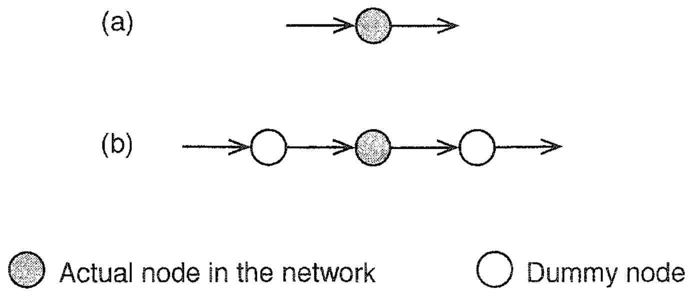


Figure 3.6 (a) A node in a network employing channel-space switching. (b) The node is modeled with two dummy nodes and two additional links.

We illustrate the modeling of a network through an example. Consider a 3×3 uni-directional mesh-torus network as shown in Figure. 3.7. Assume that every link has F fibers, W wavelengths per fiber, and T time slots per wavelength. Let C denote the total number of channels in a link. For the sake of simplicity, assume that all the links have the same number of channels. Let a node

i view the links attached to it as K_i trunks with S_i channels in each.

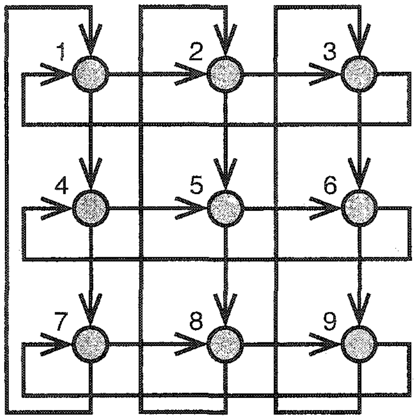


Figure 3.7 A 3×3 uni-directional mesh-torus network.

Figure. 3.8 shows modeling of the 3×3 uni-directional mesh-torus network with dummy nodes. The shaded nodes represent the actual nodes in the network while the un-shaded ones represent the dummy nodes. Let the dummy nodes around an actual nodes be labeled based on the direction (right hand side node referring to east). Let iE, iW, iN, and iS denote the dummy node connected to right, left, top, and bottom of a node i.

The actual nodes in the network are made to view links as FWT trunks and the dummy nodes that are immediate neighbors of a node i view the link as K_i trunks. A link connecting a dummy node to an actual node in the network is represented by a $K_i \times C$ matrix while a link connecting an actual node to a dummy node will be represented as a $C \times K_i$ matrix. A link connecting two dummy nodes will be represented as a $K_i \times K_j$ matrix, where K_i and K_j denote the number of trunks as viewed by the source and destination dummy nodes.

For example, assume that node 1 is a WG node and node 2 is a TG node, viewing the links as three and two trunks, respectively. The transpose of the information matrix of the link connecting

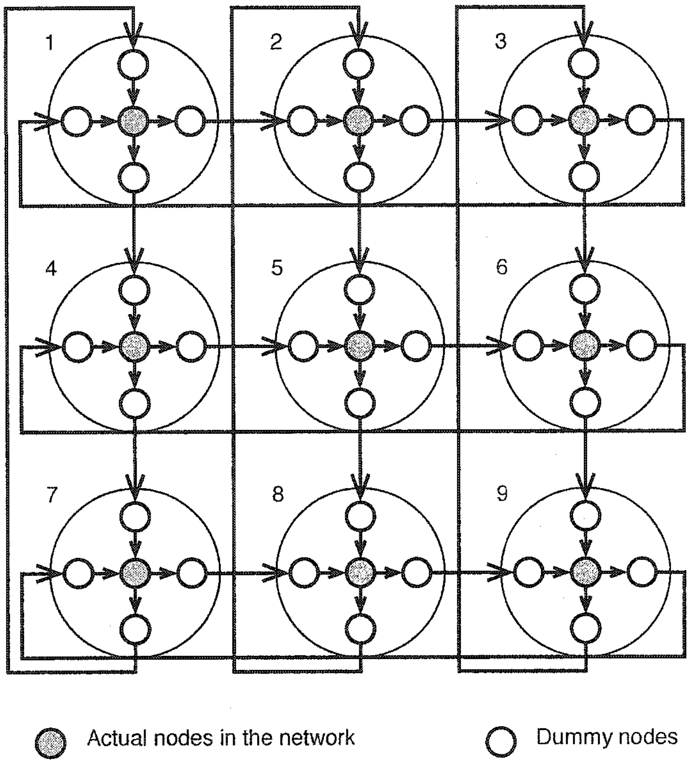


Figure 3.8 A 3×3 uni-directional mesh-torus network.

node 1 to 1E is given by[2]:

$$L_{1,1E}^{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.22}$$

The information matrix of link connecting node $2W$ to 2 is given by:

$$L_{2W,2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.23}$$

The information matrix of the link connecting node node $1E$ to $2W$ is given by:

$$L_{1E,2W} = \begin{bmatrix} 3 & 3 \\ 3 & 3 \\ 3 & 3 \end{bmatrix} \tag{3.24}$$

[2] The information matrix is represented in its transposed form to conserve space.

Note that the dummy nodes represent the full-channel interchangers that are present at the input and output stages at a node. It can be easily seen that if two nodes view the links in exactly the same manner, the link connecting the dummy node of the first link to that of the second link would be a diagonal matrix. In such a case, one of the dummy nodes can be removed from the network. If such a modified network is employed for simulation, the traffic in the network is generated and terminated only at the actual nodes in the network, hence dummy nodes do not contribute to the network traffic.

3.5 Summary

In this chapter, we developed a framework for channel establishment, called MICRON (Methodology for Information Collection and Routing in Optical Networks), in a WDM grooming network with heterogeneous switching architectures. We illustrate with examples the various information that could be collected from the links and various operators that could be used to obtain information on a path. These information can be used to select a path dynamically depending on the network status. We complete the framework by providing a generic channel assignment procedure that could be employed to implement different specific schemes. The framework can be implemented with simple traffic engineering extensions to the already existing routing protocols in the wide-area networks.

CHAPTER 4 Analysis for Blocking Performance

One of the important performance metric by which a wide-area network is evaluated is based on the success ratio of the number of requests that are accepted in the network. This metric is usually posed in its alternate form as *blocking probability* that refers to the rejection ratio of the requests in the network. The smaller the rejection ratio, the better the network performance. Although other performance metrics exist, such as effective carried traffic in the network, fairness of request rejections with respect to requests requiring different capacity requirements or different path lengths, the most meaningful way to measure the performance of a wide-area network is through blocking performance. The other performance metrics described above can, to some extent, be obtained as functions of blocking performance.

Analytical models that evaluate the blocking performance of wide-area circuit-switched networks are employed during the design phase of a network. These models are typically employed as an *elimination test*, rather than as an acceptance test, in the design phase. In other words, the analytical models are employed as back of the envelope calculations to evaluate a network design, rejecting those designs that are below a certain threshold.

In this chapter, we develop an analytical framework to evaluate the blocking performance of TSNs. The chapter is organized as below: Section 4.1 describes the assumptions on the network and traffic models considered for analysis. The estimation of call arrival rates on a link is outlined in Section 4.2. The path blocking performance is derived in Section 4.3. The computation of trunk occupancy distributions that maps the channel distribution at input and output links at a node to equivalent trunk distributions is described in Section 4.4. Analytical model for evaluating the blocking performance of a tree in a homogeneous TSN is developed by splitting a tree into multiple paths and employing the path blocking performance, described in Section 4.6. Analysis for TSNs with heterogeneous switching architectures is developed in Section 4.7.

4.1 Assumptions

We use the following assumptions to develop an analytical model for evaluating the blocking performance of a TSN.

- The network has N nodes.
- The call arrival at every node follows a Poisson process with rate λ_n. The choice of Poisson traffic is to keep the analysis tractable.
- The calls can be either unicast connections with one destination or multicast connections with more than one destination. The traffic due to multicast connections is negligible. Hence, for the derivation of path and tree blocking probabilities, the network load is assumed to be entirely due to unicast connections.
- The probability that a call requires for capacity b and is destined to a node that has a distance of z hop lengths is $p_{z,b}$. Note that the distance refers to the length of the connection that needs to be established.
- The maxmimum bandwidth requirement of a call is B and the maximum path length in the network is $N - 1$.
- The path selection is pre-determined (fixed-path routing), eg: shortest-path.
- A request cannot be handled by multiple trunks at a node.
- The holding time of every call follows an exponential distribution with mean $\frac{1}{\mu}$. The Erlang load offered by a node is $\rho_n = \frac{\lambda_n}{\mu}$.
- Blocked calls are not re-attempted.

- A call is assigned a channel randomly from a set of available channels in a sub-trunk on a link.

4.2 Estimation of call arrival rates on a link

Typically, the network traffic is specified in terms of the offered load between node pairs. The call arrival rates at the nodes have to be translated into arrival rates at individual links in the network. The computation of blocking probability depends on the link arrival rates, and the link arrival rates, in turn, depend on the network blocking probability. However, if the blocking probability in the network is small, then its effect on the link arrival rates can be ignored.

Consider a network with N nodes and L links, the average path length of a connection in the network is given by:

$$Z_{av} = \sum_{z=1}^{N-1} z\, p_z \tag{4.1}$$

where p_z is the path-length distribution. The path-length distribution is obtained from the joint probability distribution of path length and call capacity as:

$$p_z = \sum_{b=1}^{B} p_{z,b} \tag{4.2}$$

Similarly, the average capacity requirement of a connection B_{av} is given by

$$B_{av} = \sum_{b=1}^{B} b\, p_b \tag{4.3}$$

where p_b is the probability that a call requires a bandwidth of b channels and is computed as:

$$p_b = \sum_{z=1}^{N-1} p_{z,b} \tag{4.4}$$

The average resource required by a call is computed as:

$$R_{av} = \sum_{z=1}^{N-1} \sum_{b=1}^{B} zbp_{z,b} \tag{4.5}$$

Recall that λ_n denotes the call arrival rate at a node. Let λ denote the average link arrival rate and is computed as:

$$\lambda = \frac{N\lambda_n \sum_{z=1}^{N-1} \sum_{b=1}^{B} zbp_{z,b}}{L} \tag{4.6}$$

The fraction of traffic that is not destined for a node is obtained as the ratio of the number of links a path that are not the last hop to the total number of links in the path. For a path with z links, there are $(z-1)$ intermediate links. Let δ_c denote the fraction of traffic on a link that would continue on any neighboring links at a node. δ_c is computed as:

$$\begin{aligned} \delta_c &= \frac{\sum_{z=1}^{N-1} \sum_{b=1}^{B} b(z-1)p_{z,b}}{\sum_{z=1}^{N-1} \sum_{b=1}^{B} zbp_{z,b}} & (4.7) \\ &= 1 - \frac{B_{av}}{R_{av}} & (4.8) \end{aligned}$$

It is to be noted that the above expression gives the fraction of the traffic that is not destined for a node. Such traffic could continue on any of the output links at the node. The link load correlation is defined as the probability that a call on a link would continue to a successive link on a chosen path and is given by:

$$\gamma_c = \left(1 - \frac{B_{av}}{R_{av}}\right) \frac{1}{E} \tag{4.9}$$

where E denotes the number of links at the node that do not connect the node to any of the previous nodes in the path, referred to as *exit links*. Hence, the arrival rate of traffic on a link that would continue to a successive link on a path is given by $\lambda_c = \gamma_c \lambda$.

If the capacity requirement of every call in the network is one channel, then the link load correlation reduces to that derived in [18]:

$$\gamma_c = \left(1 - \frac{1}{Z_{av}}\right) \frac{1}{E} \tag{4.10}$$

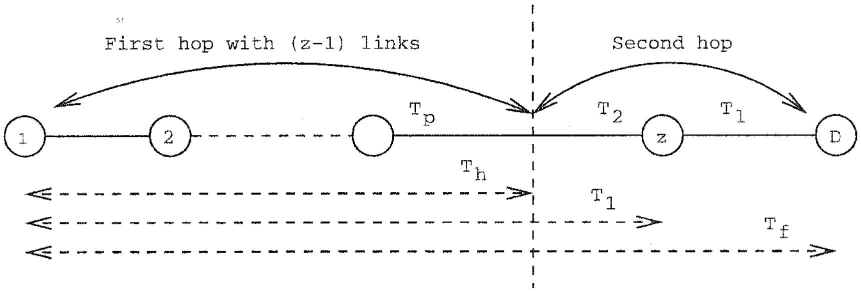


Figure 4.1 A z-link path.

4.3 Path blocking performance

The network blocking probability is computed as the average blocking probability experienced over different path lengths. Consider a z-link path model as shown in Figure 4.1. The analysis that is developed in this section assumes that the capacity requirement of a call is r channels.

Let $P_z(T_f)$ denote the probability of T_f trunks being available on a z-link path as viewed by the last node on the path[1] (node z). The definition of the trunk is as viewed by the node denoted by the suffix for P. $P_z(T_f = 0)$ denotes the blocking probability over the z-link path.

Let $P_z(T_f, T_l)$ denote the probability of T_f trunks being available on a z-link path with T_l trunks free on the last link. It can be seen that the last link should have at-least T_f trunks free, therefore $T_l \geq T_f$. $P_z(T_f)$ can then be written as:

$$P_z(T_f) = \sum_{T_l=T_f}^{K_z} P_z(T_f, T_l) \tag{4.11}$$

where K_z denotes the number trunks in the link as viewed by node z.

A z-link path is analyzed as a two-hop path by considering the first $z-1$ links as the first hop and the last two links as the second hop, as shown in Figure 4.1. Let T_h and T_p denote the number of trunks available on the first hop and that which are free on the last link of the first hop (link $z-1$), respectively, as viewed by the last node on the first hop (node $z-1$). Let T_1 and T_2 denote the number of trunks free on the first hop and number of trunks free on the last link of the first hop as seen by the node in the second hop (node z). $P_z(T_f, T_l)$ can then be recursively computed as:

[1] The destination is not considered as the last node in the path.

$$P_z(T_f, T_l) = \sum_{T_h=0}^{K_{z-1}} \sum_{T_p=T_h}^{K_{z-1}} \sum_{T_1=T_f}^{K_z} \sum_{T_2=T_1}^{K_z} P_{z-1}(T_1, T_2)\, P_{z,z-1}(T_h, T_p|T_1, T_2)\, P_z(T_f, T_l|T_h, T_p) \quad (4.12)$$

where $P_z(T_f, T_l|T_h, T_p)$ denotes the probability of T_f trunks being available on the second hop with T_l trunks free on the last link of the second hop given that T_g trunks are available on the first hop with T_p trunks free at the input to the node on the second hop. $P_{z,z-1}(T_h, T_p|T_1, T_2)$ denotes the probability that the number of trunks available on the first hop and number of trunks free on the last link of the first hop as viewed by the node in the second hop (node z) are T_h and T_p, respectively, given that the trunk availability as viewed by the last node (node $z-1$) on the first hop is T_1 and T_2. For homogeneous TSNs, for any two successive nodes $z-1$ and z on a path $P_{z,z-1}(T_h, T_p|T_1, T_2)$ is defined as:

$$P_{z,z-1}(T_h, T_p|T_1, T_2) = \begin{cases} 1 & \text{if } T_h = T_1 \text{ and } T_p = T_2 \\ 0 & \text{otherwise} \end{cases} \quad (4.13)$$

For a homogeneous TSN, Eqn. (4.12), therefore, reduces to:

$$P_z(T_f, T_l) = \sum_{T_h=T_f}^{K} \sum_{T_p=T_h}^{K} P_{z-1}(T_h, T_p)\, P(T_f, T_l|T_h, T_p) \quad (4.14)$$

where K denotes the number of trunks in a link as viewed by the nodes in the network.

The starting point of the recursion, for $z = 1$, is defined as:

$$P_1(T_f, T_l) = \begin{cases} P(T_l) & \text{if } T_f = T_l \\ 0 & \text{otherwise} \end{cases} \quad (4.15)$$

where $P(T_l)$ denotes the probability of T_l trunks being free on a link. The computation of $P(T_l)$ is discussed in Section 4.4.

$P_z(T_f, T_l|T_h, T_p)$ is computed by conditioning on the number of trunks free on the last link as viewed by node z. From this point on, we concentrate on the second hop, which is a two-link path. The definition of trunk will be assumed to be as the one viewed by the intermediate node in the two-link path.

$P_z(T_f, T_l|T_h, T_p)$ is computed as:

$$P_z(T_f, T_l|T_h, T_p) = \begin{cases} P_z(T_f|T_h, T_p, T_l)\ P_z(T_l|T_h, T_p) & \text{if } T_h \geq T_f \\ \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

where $P_z(T_l|T_h, T_p)$ denotes the probability of T_l trunks being free on the last link given that T_h trunks are available on the first hop with T_p trunks free on the last link of the first hop as viewed by node z. The number of trunks free on the last link depends on the number of trunks free on the previous links. If the correlation of traffic on a link is assumed to be only due to its previous link, then it is referred to as the *Markovian correlation*. With the assumption of Markovian correlation, $P_z(T_l|T_h, T_p)$ can be reduced to $P_z(T_l|T_p)$. Hence, Eqn. (4.16) can be written as:

$$P_z(T_f, T_l|T_h, T_p) = \begin{cases} P_z(T_f|T_h, T_p, T_l)\ P_z(T_l|T_p) & \text{if } T_h \geq T_f \\ \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

$P_z(T_f|T_h, T_p, T_l)$ denotes the probability that T_f trunks are available on the two-hop path given that T_l trunks are free on the last link and T_h trunks are available on the first hop with T_p trunks free on the last link of the first hop. $P_z(T_f|T_h, T_p, T_l)$ is computed by considering a two-link path as shown in Figure 4.2. The number of trunks free on the first link and that which are free on the second link are denoted by T_p and T_l, respectively.

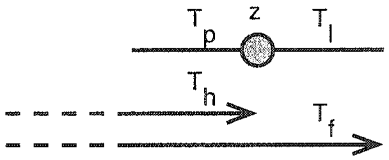


Figure 4.2 Two link path model with the free and available trunks as viewed by the intermediate node.

The trunk on a two-link path can be in any one of the following four states, as shown Figure 4.3:

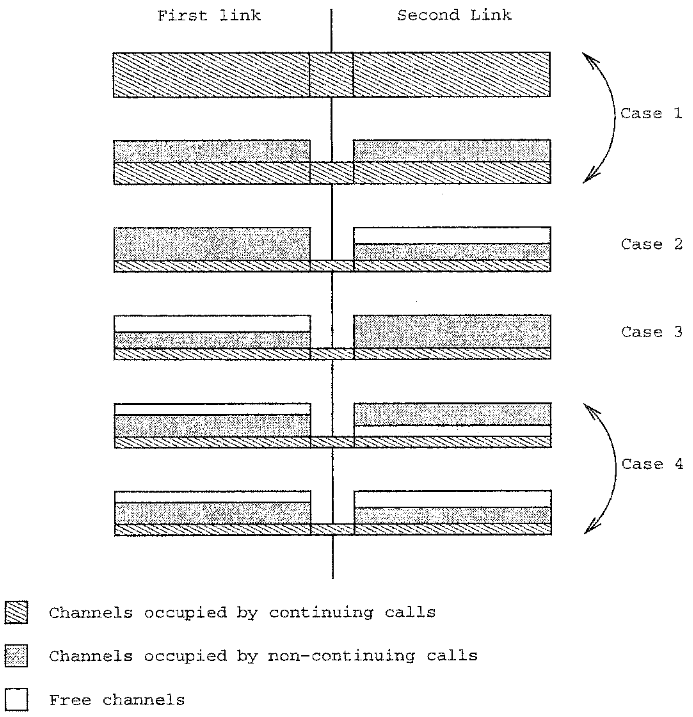

First link
Second Link
Case 1
Case 2
Case 3
Case 4
Channels occupied by continuing calls
Channels occupied by non-continuing calls
Free channels

Figure 4.3 Different possible states for trunk occupancy on a two-link path.

- **Case 1:** The trunk is busy on both the links. The trunk can be either partially or fully occupied by continuing calls. Let V_c denote the number of trunks busy on both the links.
- **Case 2:** The trunk is busy on the first link but not on the second.
- **Case 3:** The trunk is busy on the second link but not on the first.
- **Case 4:** The trunk is free on both the links. Let T_b denote the number of trunks free on both the first and second links. However, this does not imply that these trunks are available on the two-link path. Let T_a ($T_a \leq T_b$) denote the number of trunks available on the two-link path. When the node connecting the two links employs a full-permutation switch, a trunk is available on the two link path if it is free on both the links. Hence, $T_a = T_b$.

Let $P_z(T_a, T_b|T_p, T_l)$ denote the probability that T_b trunks are free on both the first and second link with T_a among them being available on the two-link path given that T_p and T_l trunks are free

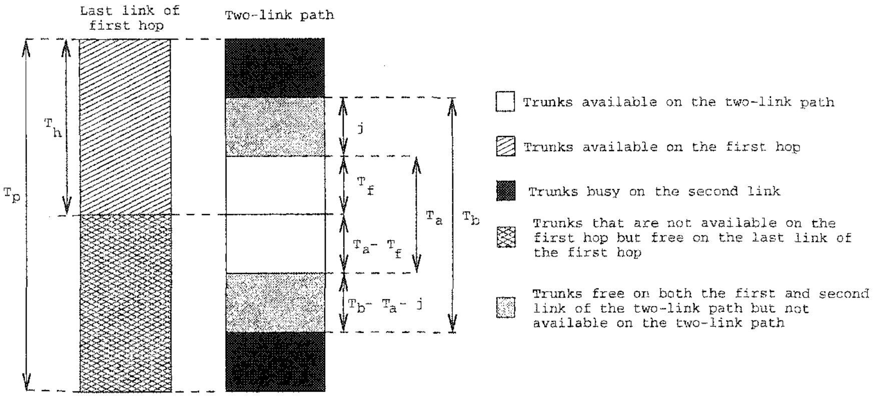

Last link of first hop
Two-link path
Tp
Th
j
Tf
Ta- Tf
Tb- Ta- j
Ta
Tb
□ Trunks available on the two-link path
▨ Trunks available on the first hop
■ Trunks busy on the second link
▩ Trunks that are not available on the first hop but free on the last link of the first hop
░ Trunks free on both the first and second link of the two-link path but not available on the two-link path

Figure 4.4 Arrangement of trunk distribution on a two-link path.

on the first and second links, respectively. $P_z(T_f|T_h, T_p, T_l)$ can then be written as:

$$P_z(T_f|T_h, T_p, T_l) = \sum_{T_a=T_f}^{min(T_p,T_l)} \sum_{T_b=T_a}^{min(T_p,T_l)} P_z(T_f|T_a, T_b, T_h, T_p, T_l)\ P_z(T_a, T_b|T_p, T_l) \tag{4.18}$$

where $P_z(T_f|T_a, T_b, T_h, T_p, T_l)$ denotes the probability of T_f trunks being available on the two-hop path given that T_h trunks are available on the first hop, T_p trunks are free on the first link (the first link in the two-link model is the last link on the first hop), T_l trunks are free on the second link, T_b trunks are free on both the first and second link and T_a among them available on the two-link path.

Figure 4.4 shows the pictorial view of the distribution of free trunks on the last link of the first hop and that of the two-link path model. From this figure, $P_z(T_f|T_a, T_b, T_h, T_p, T_l)$ can be computed along the lines of the following argument.

Assume that T_p trunks are free on the last link of the first hop with T_h among them available on the first hop. Also assume that T_b trunks are free on the first and second link of the two-link path with T_a among them being available on the two-link path. Among the T_a available trunks on the two-link path exactly T_f trunks overlap with the T_h trunks available on the first hop. The remaining trunks that are available on the first hop, $T_h - T_f$, are not available on the two-link path. This could occur in two cases: (1) the corresponding trunk is busy on the second link of the two-link path, or (2) the trunk is free on the second link but is not available on the two-link path (due to switching constraints). The number of trunks that satisfy the latter case is $T_b - T_a$. The required probability is computed by assuming that j of the trunks that are free on both the first and second link, but not available on the two-link path, overlap with the remaining $T_h - T_f$ available trunks of the first hop. The trunks on the second link corresponding to the remaining $T_h - T_f - j$ available trunks on the first-hop are busy. Thus, $P_z(T_f|T_a, T_b, T_h, T_p, T_l)$ can be written as:

$$P_z(T_f|T_a, T_b, T_h, T_p, T_l) = \frac{\binom{T_h}{T_f}\binom{T_p-T_h}{T_a-T_f}\sum_j \binom{T_h-T_f}{j}\binom{T_p-T_h-T_a+T_f}{T_b-T_a-j}}{\binom{T_p}{T_b}\binom{T_b}{T_a}} \tag{4.19}$$

where $max(0, T_b + T_h - T_p - T_f) \leq j \leq min(T_h - T_f, T_b - T_a)$. For the special case, when the switch at a node has full-permutation switching capability, the above equation reduces to:

$$P_z(T_f|T_a, T_b, T_h, T_p, T_l) = \begin{cases} \frac{\binom{T_h}{T_f}\binom{T_p-T_h}{T_a-T_f}}{\binom{T_p}{T_a}} & \text{if } T_a - T_f \leq T_p - T_h \\ \\ 0 & \text{otherwise.} \end{cases} \tag{4.20}$$

Also, for this case, $P_z(T_a, T_b|T_p, T_l) = 0$ if $T_a \neq T_b$. Hence, Eqn. (4.18) can be written as:

$$P_z(T_f|T_h, T_p, T_l) = \sum_{T_a=T_f}^{min(T_p,T_l)} P(T_f|T_a, T_a, T_h, T_p, T_l)\ P(T_a, T_a|T_p, T_l) \tag{4.21}$$

The probability values, $P_z(T_a, T_b|T_p, T_l)$, $P_z(T_l|T_p)$, and $P_z(T_l)$ are computed by considering a switch model as explained in the following subsections.

4.4 Free trunk distribution

Consider a two-link path model as shown in Figure 4.5. Let u_p, u_l, and u_c denote the number of channels busy on the first link, number of channels busy on second link, and number of channels occupied by calls that continue from the first link to the second, respectively. Note that $u_c \leq min(u_p, u_l)$.

The number of channels busy on a trunk at the input of node z is the same as the number of channels busy at the input of the switch (after the FCI) at the node, while the distribution of the

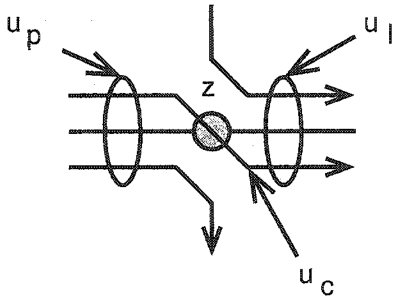


Figure 4.5 Two link model with channel distribution.

busy channels at the input of the switch is independent of the distribution at the input of the node. Also, the state of a trunk (busy or free) at the input of the node is the same as that at the input to the switch. Therefore, the first link as referred to in this subsection would henceforth correspond to the link viewed at the input of the switch.

Let $\lambda_p^{(b)}$, $\lambda_l^{(b)}$, and $\lambda_c^{(b)}$ denote the arrival rate for calls that request for capacity b to the first link, the second link, and those that continue from the first link to the second. Note that $\lambda_c^{(b)} \leq min(\lambda_p^{(b)}, \lambda_l^{(b)})$. The Erlang loads corresponding to the calls that occupy the first link, second link, and that which continue from the first to the second can be written as, $\rho_p^{(b)} = \frac{\lambda_p^{(b)}}{\mu}$, $\rho_l^{(b)} = \frac{\lambda_l^{(b)}}{\mu}$, and $\rho_c^{(b)} = \frac{\lambda_c^{(b)}}{\mu}$, respectively.

Let u_p, u_l, and u_c denote the number of channels busy on the first link, number of channels busy on second link, and number of channels occupied by calls that continue from the first link to the second, respectively. Note that $u_c \leq min(u_p, u_l)$.

The channel distribution on a two-link path can be characterized to a limited extent as a 3-dimensional Markov chain. The state-space is denoted by the 3-tuple (u_p, u_l, u_c). Note that such a representation of the state of the two-links does not take into account the number of calls that require a certain specified bandwidth. The steady-state probability for the states is computed recursively by considering the number of calls that require a certain capacity, with B being the maximum capacity requirement of a connection, as:

$$\Pi(u_p, u_l, u_c) = \frac{1}{G}\xi_B(u_p, u_l, u_c) \tag{4.22}$$

where $0 \leq u_p \leq KS$, $0 \leq u_l \leq KS$, and $0 \leq u_c \leq min(u_p, u_l)$. G is the normalization constant and is defined as:

$$G = \sum_{u_c=0}^{KS} \sum_{u_p=0}^{KS} \sum_{u_l=0}^{KS} \xi_B(u_p, u_l, u_c) \tag{4.23}$$

$\xi_B(u_p, u_l, u_c)$ is recursively computed as:

$$\xi_b(u_p, u_l, u_c) = \begin{cases} \sum_{z=0}^{\lfloor \frac{u_c}{b} \rfloor} \sum_{x=z}^{\lfloor \frac{u_p}{b} \rfloor} \sum_{y=z}^{\lfloor \frac{u_l}{b} \rfloor} \\ \frac{(\rho_p^{(b)} - \rho_c^{(b)})^{x-z}}{(x-z)!} \frac{(\rho_c^{(b)})^z}{z!} \frac{(\rho_p^{(b)} - \rho_c^{(b)})^{y-z}}{(y-z)!} \xi_{b-1}(u_p - bx, u_l - by, u_c - bz) & \text{if } b > 1 \\ \frac{(\rho_p^{(b)} - \rho_c^{(b)})^{u_p-u_c}}{(u_p-u_c)!} \frac{(\rho_c^{(b)})^{u_c}}{u_c!} \frac{(\rho_p^{(b)} - \rho_c^{(b)})^{u_l-u_c}}{(u_l-u_c)!} & \text{if } b = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.24}$$

Let V_p, V_l, and V_c denote the number of trunks busy on the first link, number of trunks busy on the second link, and number of trunks that are busy on both the first and second links, respectively. It can be observed that $V_c \leq min(V_p, V_l)$. The number of trunks free on both the links is given by, $T_b = K_z - (V_p + V_l - V_c)$. The number of trunks available on the two-link path is denoted by T_a. The state-space of the trunk distribution is captured by the 4-tuple (V_p, V_l, V_c, T_a). The steady-state probability of the states can be computed by conditioning on the channel distribution, (u_p, u_l, u_c) as:

$$\psi(V_p, V_l, V_c, T_a) = \sum_{u_c=0}^{KS} \sum_{u_p=u_c}^{KS} \sum_{u_l=u_c}^{KS} P(V_p, V_l, V_c, T_a | u_p, u_l, u_c)\, \Pi(u_p, u_l, u_c) \tag{4.25}$$

where $P_z(V_p, V_l, V_c, T_a | u_p, u_l, u_c)$ denotes the probability that the trunk distribution is in state (V_p, V_l, V_c, T_a) given that the channel distribution is (u_p, u_l, u_c). The following probability values that are required to complete the analytical model, described in the previous section, can then be derived from the above steady-state probability.

$$P_z(T_a, T_b|T_p, T_l) = \frac{\psi(K_z - T_p, K - T_l, K_z + T_b - T_p - T_l, T_a)}{\sum_{t_a=0}^{min(T_p,T_l)} \sum_{t_b=t_a}^{min(T_p,T_l)} \psi(K_z - T_p, K - T_l, K_z + t_b - T_p - T_l, t_a)} \tag{4.26}$$

$$P_z(T_l|T_p) = \frac{\sum_{t_a=0}^{min(T_p,T_l)} \sum_{t_b=0}^{min(T_p,T_l)} \psi(K_z - T_p, K - T_l, K_z + t_b - T_p - T_l, t_a)}{\sum_{t_l=0}^{K_z} \sum_{t_a=0}^{min(T_p,t_l)} \sum_{t_b=0}^{min(T_p,t_l)} \psi(K_z - T_p, K - t_l, K_z + t_b - T_p - t_l, t_a)} \tag{4.27}$$

$$P_z(T_l) = \sum_{t_p=0}^{K_z} \sum_{t_a=0}^{min(t_p,T_l)} \sum_{t_b=t_a}^{min(t_p,T_l)} \psi(K_z - t_p, K - T_l, K_z + t_b - t_p - T_l, t_a) \tag{4.28}$$

The trunk occupancy probability for a given a channel distribution, is computed as:

$$P_z(V_p, V_l, V_c, T_a|u_p, u_c, u_l) = \frac{N_{k=K_z}(V_p, V_l, V_c, T_a|u_p, u_l, u_c)}{A_{k=K_z}(u_p, u_l, u_c)} \tag{4.29}$$

where $N_k(V_p, V_l, V_c, T_a|u_p, u_l, u_c)$ denotes the number of ways of arranging across k trunks, u_p busy channels on the first link, u_l busy channels on the second link, with u_c channels among them being occupied by calls that continue from the first link to second, such that V_p trunks are busy on the first link, V_l trunks are busy on the second link with V_c among them busy on both the links, and T_a trunks being available on the two-link path. $A_k(u_p, u_l, u_c)$ denotes all possible ways of arranging across k trunks, u_p busy channels on the first link, u_l busy channels on the second link with u_c channels among them being occupied by calls that continue from the first link to second. $A_k(u_p, u_l, u_c)$ is recursively computed as:

$$A_k(u_p, u_l, u_c) = \sum_{z=0}^{min(S,u_c)} \sum_{x=z}^{min(S,u_p)} \sum_{y=z}^{min(S,u_l)} A_1(x, y, z)\ A_{k-1}(u_p - x, u_l - y, u_c - z) \tag{4.30}$$

where $0 \leq u_p \leq kS$, $0 \leq u_l \leq kS$, and $0 \leq u_c \leq min(u_p, u_l)$. The definition of $A_1(x, y, z)$ depends on the nature of switch.

$N_k(V_p, V_l, V_c, T_a|u_p, u_l, u_c)$ (written as $N_k(.)$ for short due to space constraints) is assigned 0 if any of the following conditions hold true:

- $\{V_p, V_l, V_c, T_a, u_p, u_l, u_c\} < 0$
- $\{V_p, V_l, V_c, T_a\} > k$
- $\{u_p, u_l\} > kS$ or $u_c > min(u_p, u_l)$
- $u_p < V_pS$ or $u_l < V_lS$

Otherwise, it is computed recursively under one of the following four cases, as described in Figure 4.3:

Case 1: If $V_c > 0$

The required probability is obtained by conditioning on a trunk being busy on both the links.

$$N_k(.) = \frac{k}{V_c} \sum_{z=0}^{S} A_1(S, S, z) N_{k-1}(V_p - 1, V_l - 1, V_c - 1, T_a | u_p - S, u_l - S, u_c - z) \qquad (4.31)$$

Case 2: If $V_c = 0, V_p > 0$

The required probability is obtained by conditioning on a trunk being busy on the first link but free on the second link.

$$N_k(.) = \frac{k}{V_p} \sum_{z=0}^{min(S-1,u_c)} \sum_{y=z}^{min(S-1,u_l)} A_1(S, y, z)\, N_{k-1}(V_p - 1, V_l, V_c, T_a | u_p - S, u_l - y, u_c - z) \quad (4.32)$$

Case 3: If $V_c = 0, V_p = 0, V_l > 0$

The required probability is obtained by conditioning on a trunk being free on the first link but busy on the second link.

$$N_k(.) = \frac{k}{V_l} \sum_{z=0}^{min(S-1,u_c)} \sum_{x=z}^{min(S-1,u_p)} A_1(x, S, z) N_{k-1}(V_p, V_l - 1, V_c, T_a | u_p - x, u_l - S, u_c - z) \quad (4.33)$$

Case 4: If $V_c = 0, V_p = 0, V_l = 0$

The required probability is obtained on the condition that a trunk is free on both the links. Two possible cases need to be considered: (1) the trunk is available on the two-link path or (2) the trunk is not available on the two-link path. Let $B_1(x, y, z)$ denote the number of ways of arranging on a trunk, x busy channels on the first link, y busy channels on the second link, with z channels among them being occupied by calls that continue from the first link to second, such that the trunk is not available on the two-link path. Similarly, let $F_1(x, y, z)$ denote the arrangement of the busy channels on a trunk such that the trunk is available on the two-link path. It can be observed that $F_1(x, y, z) + B_1(x, y, z) = A_1(x, y, z)$. $N_k(.)$ can then be computed as:

$$\begin{aligned} N_k(.) = & \textstyle\sum_{z=0}^{min(S-1,u_c)} \sum_{x=z}^{min(S-1,u_p)} \sum_{y=z}^{min(S-1,u_l)} \\ & [F_1(x,y,z)N_{k-1}(V_p, V_l, V_c, T_a - 1|u_p - x, u_l - y, u_c - z) \\ & +B_1(x,y,z)N_{k-1}(V_p, V_l, V_c, T_a|u_p - x, u_l - y, u_c - z)] \end{aligned} \tag{4.34}$$

The starting point of the recursion (for $k = 1$), denoted by $N_1(V_p, V_l, V_c, T_a|u_p, u_l, u_c)$, is assigned 0 if any of the following conditions hold true:

1. $V_p = 0$ and $u_p = S$
2. $V_l = 0$ and $u_l = S$
3. $V_c = 0$ and $min(u_p, u_c) = S$.

Otherwise, it is defined in terms of $B_1(u_p, u_l; u_c)$ and $F_1(u_p, u_l, u_c)$ as,

$$N_1(.) = \begin{cases} F_1(u_p, u_l, u_c) & \text{if } T_a = 1 \text{ and } V_p = V_l = V_c = 0 \\ \\ B_1(u_p, u_l, u_c) & \text{if } T_a = 0 \\ \\ 0 & \text{otherwise.} \end{cases} \tag{4.35}$$

The definitions of $A_1(u_p, u_l, u_c)$, $B_1(u_p, u_l, u_c)$, and $F_1(u_p, u_l, u_c)$ depend on the switch architecture.

4.5 Example switch models

Two kinds of switches are modeled in this section: space-only switch and full-permutation switch. For a space-only switch, channel continuity constraint is enforced by the switch. Hence, a call continuing from the first link to the second occupies the same channel at the input and output of the switch. Note that although the switch is space-only, the switching provided by the node is channel-space due to the full-channel interchanger at the input of the node. A full-permutation switch, on the other hand, can switch any free channel at the input to any free channel at the output. Consider a call that requires a capacity of b channels, for a trunk with S channels, $A_1(u_p, u_l, u_c)$ and $B_1(u_p, u_l, u_c)$ are defined as:

Space-only switch:

$$A_1(u_p, u_l, u_c) = \begin{cases} \binom{S}{u_p}\binom{u_p}{u_c}\binom{S-u_c}{u_l-u_c} & \text{if } 0 \leq u_p, u_l \leq S \\ & \text{and } u_c \leq min(u_p, u_l) \\ 0 \quad \text{otherwise.} & \end{cases} \tag{4.36}$$

$$B_1(u_p, u_l, u_c) = \begin{cases} \sum_{r=0}^{b-1}\binom{S-r}{u_p}\binom{u_p}{u_c}\binom{u_p-u_c}{S-r-u_l} & \text{if } 0 \leq u_p, u_l \leq S \\ & u_c \leq min(u_p, u_l), \text{ and} \\ & u_p + u_l - u_c \geq S - b \\ 0 \quad \text{otherwise.} & \end{cases} \tag{4.37}$$

Full-permutation switch:

$$A_1(u_p, u_l, u_c) = \begin{cases} \binom{S}{u_p}\binom{S}{u_l} & \text{if } 0 \leq u_p, u_l \leq S \\ & \text{and } u_c \leq min(u_p, u_l) \\ 0 \quad \text{otherwise.} & \end{cases} \tag{4.38}$$

$$B_1(u_p, u_l, u_c) = \begin{cases} \binom{S}{u_p}\binom{S}{u_l} & \text{if } min(S-u_p, S-ul) < b \\ & \text{and } u_c \leq min(u_p, u_l). \\ 0 \quad \text{otherwise.} & \end{cases} \tag{4.39}$$

It can be observed that the analytical models proposed earlier in the literature can be derived from this generalized model as:

- $K = W$; $S = 1$; $\gamma_c = 0$; [12, 15]
- $K = W$; $S = 1$; [18]
- $K = W$; $S = T$; $\gamma_c = 0$; full-permutation switch [9]
- $K = W$; $S = F$; full-permutation switch. [19]

4.6 Multicast tree establishment in TSNs

Supporting multicast connections in WDM networks has gained importance in recent years due to the increasing number of distributive services. The benefits of supporting multicast traffic at the WDM layer are discussed in [40]. Various multicast models are introduced in [41] along with models to implement different multicast capable switch architectures.

Earlier work on the analysis of optical multicasting concentrate on two main areas: (1) minimizing the number of wavelengths required to support a static traffic demand [40, 42] and (2) multicast route selection algorithms that provide efficient utilization of the fiber bandwidth when dynamic setup and tear down of multicast traffic is considered [43, 44]. The blocking performance for establishing trees has been studied using link-independence model in [45]. However, the study is restricted to trees where the source reaches the destinations on a two-link path with a branching after the first link. The analytical model developed in [46] for bloking performance of wavelength-routed networks under multicast trafic considers a completely connected network. The results obtained can be employed as lower bounds for other topologies. In [47], the model considers multicast and unicast traffic with multiple clases of services. It has to be noted the above mentioned models are applicable to networks employing wavelength-routing and sparse- or no wavelength conversion.

To the best of our knowledge, there has not been any work that analyzes the blocking performance of establishing multicast connections in WDM grooming networks.

4.6.1 Blocking performance of tree establishment

Consider a tree, denoted by $\mathcal{T}$, that needs to be established in a homogeneous TSN. Let $P_{\mathcal{T}}(T_f)$ denote the probability that exactly T_f trunks are available to establish the tree. $P_{\mathcal{T}}(0)$, therefore, denotes the blocking probability for tree establishment. To compute $P_{\mathcal{T}}(T_f)$, assume that T_y trunks are free on the first link and T_x trunks among them are available for establishing the tree. $P_{\mathcal{T}}(T_f)$ can be written as,

$$P_{\mathcal{T}}(T_f) = \sum_{T_x=T_f}^{K} \sum_{T_y=T_x}^{K} P_{\mathcal{T}}(T_f|T_x,T_y)P_1(T_x,T_y) \tag{4.40}$$

where $P_{\mathcal{T}}(T_f|T_x,T_y)$ denotes the probability of T_f trunks being available to establish the tree given that T_y trunks are free on the first link with T_x among them being available. $P_1(T_x,T_y)$ denotes the probability that T_y trunks are free on the first link with T_x among them being available. $P_1(T_x,T_y)$ can be written as,

$$P_1(T_x,T_y) = \begin{cases} P(T_y) & \text{if } T_x = T_y \\ 0 & \text{otherwise.} \end{cases} \tag{4.41}$$

where, $P(T_y)$ denotes the probability of T_y trunks being free on a link.

$P_{\mathcal{T}}(T_f|T_x,T_y)$ is computed by considering two cases, (1) if the tree does not have any branching and (2) if the tree has a branching.

Case 1:

If the tree $\mathcal{T}$ does not have any branching, then it is merely a path. If the path consists of z links, then $P_{\mathcal{T}}(T_f|T_x,T_y)$ reduces to $P_z(T_f|T_x,T_y)$. This probability can be expressed as:

$$P_z(T_f|T_x,T_y) \quad = \quad \sum_{T_l=T_f}^{K} P_z(T_f,T_l|T_x,T_y) \tag{4.42}$$

B where $P_z(T_f,T_l|T_x,T_y)$ denotes the probability of having T_f trunks available on a z-hop path with T_l trunks free on the last link given that T_y trunks are free on the first link with T_x among them available.

A z-link is analyzed as a two-hop path by considering the first $z-1$ links as the first hop and last two links as the second hop, as shown in Figure 4.1.

Let T_h and T_p denote the number of trunks available on the first hop and that which are free on the last link of the first hop (link $z-1$). $P_z(T_f,T_l|T_x,T_y)$ can then be recursively computed as:

$$P_z(T_f,T_l|T_x,T_y) = \sum_{T_h=T_f}^{K} \sum_{T_p=T_h}^{K} P_{z-1}(T_h,T_p|T_x,T_y) \quad P(T_f,T_l|T_h,T_p) \tag{4.43}$$

where $P(T_f,T_l|T_h,T_p)$ denotes the probability of T_f trunks being available on the z-link path with T_l trunks free on the last link given that T_h trunks are available on the first hop with T_p trunks free on the last link of the first hop. It can be observed that this probability involves only the last two links, hence is computed using a two-link model as described in Section ??.

The starting point of the recursion $P_1(T_f,T_l|T_x,T_y)$ is given by:

$$P_1(T_f,T_l|T_x,T_y) = \begin{cases} 1 & \text{if } T_f = T_x \text{ and } T_l = T_y \\ 0 & \text{otherwise.} \end{cases} \tag{4.44}$$

Case 2:

If the tree branches at an intermediate node, then the computation of the desired probability is carried out by splitting the tree into a combination of a path and subtrees. Consider an example tree as shown in Figure 4.6(a) to be established. As the tree branches, it is split into a path up to the intermediate node I, denoted by $\mathcal{P}$, and a set of subtrees that branch out at the intermediate node. Let s denote the number of subtrees at the branching point and $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_s$ denote the subtrees.

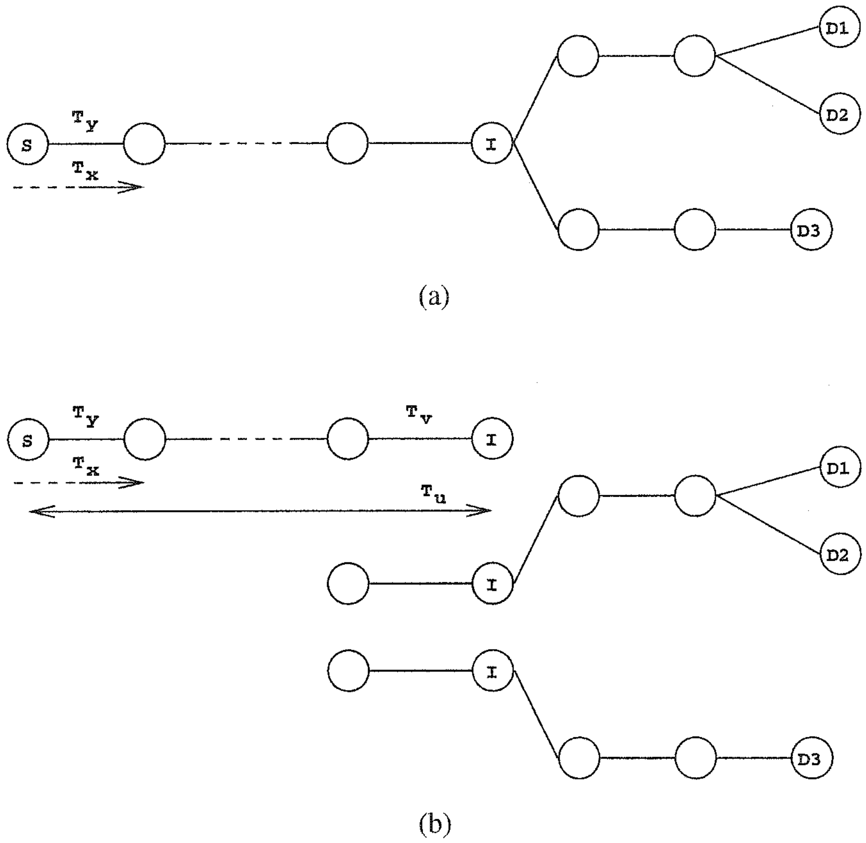


Figure 4.6 (a) Example multicast tree considered for analysis. (b) Decomposition of the tree into a path and a set of sub-trees for analysis.

The splitting of the tree into a path and a set of subtrees are shown in Figure 4.6(b). Note that the last link of the path and first link of the subtrees are the same.

Let $P_{\mathcal{P}}(T_u, T_v|T_x, T_y)$ denote the probability that T_u trunks are available to reach the intermediate node with T_v trunks free on the last link of the path given that the first link has T_y free trunks with T_x among them being available. The probability of T_f trunks being available to establish the given tree can be obtained by summing over all possible values of T_u, T_v, and number of trunks available to establish paths on each of the s subtrees. Hence, it follows:

$$P_{\mathcal{T}}(T_f|T_x, T_y) = \sum_{T_u=T_f}^{T_x} \sum_{T_v=T_u}^{K} \sum_{(t_1,t_2,...,t_s)} P_{\mathcal{P}}(T_u, T_v|T_x, T_y)\, P(t_1, t_2, ..., t_s|T_u, T_v) P(T_f|t_1, t_2, ..., t_s, T_u) \tag{4.45}$$

where $P(t_1, t_2, ..., t_s|T_u, T_v)$ denotes the joint probability that subtree i has exactly t_i trunks available given that the first link in the subtree has T_v trunks free with T_u among them available. $P(T_f|t_1, t_2, ..., t_s, T_u)$ denotes the probability of T_f trunks being available for establishing the tree given that T_u trunks are available to establish the path and t_i trunks are available to establish subtree $\mathcal{T}_i$. Note that this probability does not depend on the number of free trunks on the first link of the subtrees (T_v) as any trunk that is available to establish a subtree must be within the set of available trunks in the first link of the subtree (T_u). It can be observed that $t_i \geq T_f, 1 \leq i \leq s$.

It is assumed that the distribution of the channels across different subtrees are independent of one another. Similar to link correlation, there is also a correlation factor that is introduced due to the intra-channel copying. Hence, if a channel is occupied in a subtree, then there is a positive probability that a channel in the same trunk can be occupied in another subtree, as they could be part of a tree established earlier. However, as the offered load due to multicast traffic is expected to be much smaller compared to the unicast connections, this correlation is neglected in this paper. Assuming the channel distribution across the subtrees are independent of each other, $P(t_1, t_2, ..., t_s|T_u, T_v)$ can be written as:

$$P(t_1, t_2, ..., t_s|T_u, T_v) = \prod_{i=1}^{s} P_{\mathcal{T}_i}(t_i|T_u, T_v) \tag{4.46}$$

Let $P_s(T_f|t_1, t_2, ..., t_s, T_u, T_r)$ denote the probability of T_f trunks being available to establish the tree given that t_i trunks are available to establish subtree T_i with $T_u + T_r$ trunks available on the first link of the subtrees with the constraint that a trunk that is available to establish the tree must fall within the T_u set of trunks. It follows that $P_s(T_f|t_1, t_2, ..., t_s, T_u)$ is the same as $P_s(T_f|t_1, t_2, ..., t_s, T_u, 0)$. $P_s(T_f|t_1, t_2, ..., t_s, T_u, T_r)$ is computed recursively by considering one subtree at a time and updating the number of trunks available to establish the tree depending on the number of available trunks on the subtree that is considered.

$$P_s(T_f|t_1, t_2, ..., t_s, T_u, T_r) = \sum_{t=T_f}^{\min(t_1,T_u)} P_1(t|t_1, T_u, T_r)\; P_{s-1}(T_f|t_2, t_3, ..., t_s, t, T_r + T_u - t) \quad (4.47)$$

where,

$$P_1(t|t_1, T_u, T_r) = \frac{\binom{T_u}{t}\binom{T_r}{t_1-t}}{\binom{T_u+T_r}{t_1}}. \quad (4.48)$$

$P(T_f)$ and $P(T_f, T_l|T_h, T_p)$ form the basis for the analysis developed in this section. These probabilities are computed using a two-link correlation model as described in Section 4.4.

4.7 Mapping of trunk proabilities for heterogeneous switch architectures

In order to analyze networks with heterogeneous node architectures, the mapping of the trunk distributions from one node architecture to the other has to be computed.

Consider the intermediate link of a two-hop path connected by two nodes with different swtiching architectures as shown in Figure 4.7.

The first node views the link as K_1 trunks with S_1 channels per trunk while the second node views the link as K_2 trunks with S_2 channels per trunk. Let T_1 denote the number of available trunks on the first hop with T_2 denote the number T_p trunks free on the last link of the first hop as viewed by the last node of the first hop. Let T_1 denote the number of available trunks on the first hop and T_2 denote the number of free trunks on the last link of the first hop as viewed by the first

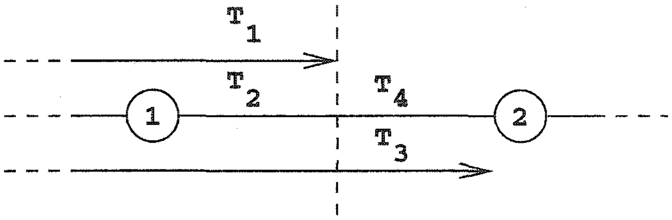


Figure 4.7 Trunk distribution of a link as viewed by two nodes with different switching architectures.

node in the second hop. Let f_2 denote the number of available channels[2] and f_1 denote the number of free channels on the link.

The required mapping, $P(T_3, T_4|T_1, T_2)$ is then computed as:

$$P(T_3, T_4|T_1, T_2) = \sum_{f_1=0}^{KS} \sum_{f_2=f_1}^{KS} P(f_1, f_2|T_1, T_2) P(T_3, T_4|f_1, f_2) \tag{4.49}$$

where $P(f_1, f_2|T_1, T_2)$ denotes the probability of f_2 channels being free on the link with f_1 among them being available given that T_2 trunks are free on the link with T_1 among them being available for a path upto that link. $P(T_3, T_4|f_1, f_2)$ denotes the probability of having T_4 free trunks with T_3 among them available as viewed by another node given that f_2 channels are free on the link with f_1 among them being available.

$P(f_1, f_2|T_1, T_2)$ is computed by conditioning on the number of free channels available in the link as:

$$\begin{aligned} P(f_1, f_2|T_1, T_2) &= P(f_2|T_1, T_2)\, P(f_1|T_1, T_2, f_1) && (4.50) \\ &= P(f_2|T_2)\, P(f_1|T_1, T_2, f_1) && (4.51) \end{aligned}$$

where $P(f_2|T_1, T_2)$ denotes the probability of having f_2 channels free in the link given that T_2 trunks are free with T_1 among them being available. This is reduced to $P(f_2|T_2)$ as the number of free channels on the link does not depend on the available trunks in the path. This probability is computed using the two-link model described in earlier sections.

[2] Available channels are those free channels in the set of available trunks.

$P(f_1|T_1, T_2, f_2)$ denotes the probability of f_1 channels being available on the link given that T_2 trunks are free with T_2 of them being available and f_2 channels free. It is computed as:

$$P(f_1|T_1, T_2, f_2) = \frac{\xi_{T_1}(f_1)\xi_{T_2-T_1}(f_2 - f_1)}{\sum_{f=0}^{f_2} \xi_{T_1}(f)\xi_{T_2-T_1}(f_2 - f)} \tag{4.52}$$

where $\xi_t(f)$ denotes the number of ways of arranging f free (or available) channels across t free (or available) trunks such that each trunk has atleast one free (or available) channel in it. $\xi_t(f)$ is computed as:

$$\xi_T(f) = \sum_{x=1}^{min(S_1,f)} \xi_1(x)\xi_{t-1}(f - x) \tag{4.53}$$

$\xi_1(x)$ denotes the number of ways of arranging x free channels over a trunk. With S_1 channels per trunk, $\xi_1(x)$ is written as:

$$\xi_1(x) = \begin{cases} \binom{S_1}{x} & \text{if } 1 \leq x \leq S_1 \\ \\ 0 & \text{otherwise.} \end{cases} \tag{4.54}$$

The probability of finding the trunk distribution as viewed by the second node given the trunk distribution and the channel distribution as viewed by the first node depends on how the channels are distributed across the trunks at the two nodes. While there could be several possible choices, there are two cases that are of interest: (1) only the number of trunks that a link is viewed as is known for both the nodes and the exact architectures are not known; and (2) the precise grooming architecture at the two nodes are known.

Case 1: Architecture independent mapping

In this case, we assume the the exact architecture of the two nodes connected to the link are not known. The only information that is known is the number of trunks that each node views the link as. The precise mapping of channels from a trunk as viewed by one node to that viewed by the other is not known. Due to the lack of the exact architecture, the knowledge of the channel distribution alone is employed for mapping the trunk distribution. The required probability,

$P(T_3, T_4|T_1, T_2, f_1, f_2)$ is computed as:

$$P(T_3, T_4|T_1, T_2, f_1, f_2) = \begin{cases} \frac{\zeta_{T_4}(T_3, f_1, f_2)}{\sum_{t_3=0}^{K_2} \sum_{t_4=t_3}^{K_2} \zeta_{t_4}(t_3, f_1, f_2)} & \text{if } T_3 \leq T_4 \\ 0 & \text{otherwise} \end{cases} \tag{4.55}$$

where $\zeta_t(t', f_1, f_2)$ denotes the number of ways of arranging f_2 calls across t trunks such that a trunk having a call belonging to the f_1 available trunk would result in exactly t' trunks being available. This value is computed as:

$$\zeta_t(t', f_1, f_2) = \begin{cases} \frac{t}{t'} \sum_{x=1}^{min(f_1, S_2)} \sum_{y=x}^{min(f_2, S_2)} \\ \quad \zeta_{t-1}(t'-1, u_1 - x, u_2 - y)\zeta_1(1, x, y) \\ \quad \text{if } t' > 0 \text{ and } u_1 > 0 \\ \sum_{y=1}^{min(f_2, S_2)} \zeta_{t-1}(t, f_1, f_2 - y)\zeta_1(1, x, y) \\ \quad \text{if } t' = 0 \text{ and } u_1 = 0 \end{cases} \tag{4.56}$$

It has to be noted that the computation of the probability $P(T_3, T_4|T_1, T_2, f_1, f_2)$ does not depend on T_1 and T_2.

Case 2: Architecture-dependent mapping

This choice arises from the architectural viewpoint. Note that when a link has multiple fibers, wavelengths and time slots, the alternatives for trunk switching are limited to treating either wavelength or time slot as a trunk, when only limited switching is allowed. In such a case, two nodes that view the link differently would have the channel distribution as considered here. For example, consider a link with two wavelengths and three time slots per wavelength. Let Node 1 view the link as wavelength trunks, i.e., 2 trunks with three channels in each. Let Node 2 view the link as time slot trunks, i.e., 3 trunks with 2 channels in each. This scenario is depicted in Fig. 4.8 showing the distribution of channels across trunks as seen by the two nodes.

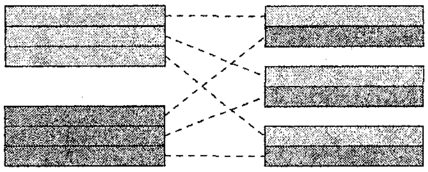


Figure 4.8 Channel distribution across trunks as viewed by two nodes employing different switching architectures.

In this case, due to the regularity in the channel distribution, the knowledge of the trunk and channel distribution as seen by Node 1 could be used to derive the lower bound on the trunk distribution as seen by Node 2. For example, if three channels are free with one trunk being free as viewed by Node 1, then, the a minimum of three trunks need to be free as viewed by Node 2. In general, if f_1 channels and T_1 ($f_1 > 0$ and $T_1 > 0$) trunks are free, then a minimum of $\frac{f_1 K_2}{T_1 S_1}$ trunks must be free as viewed by the second node. Recall that, K_2 denotes the total number of trunks in a link as viewed by Node 2 and S_1 denotes the number of channels per trunk as viewed by Node 1. The same reasoning is true for the lower bound on the available trunks as well.

The required probability $P(T_3, T_4|T_1, T_2, f_1, f_2)$ is then computed by setting the probability values of those trunk distributions that are not feasible to zero, specifically $P(T_3, T_4|T_1, T_2, f_1, f_2)$ is set to 0 if one of the following holds true.

$$T_1 > 0 \text{ and } T_3 < \frac{f_1 K_2}{T_1 S_1} \tag{4.57}$$

$$T_2 > 0 \text{ and } T_4 < \frac{f_2 K_2}{T_2 S_1} \tag{4.58}$$

The probabilities are then normalized to obtain the sum of all the conditional probabilities to 1. These pruning of state-space depends entirely on the architecture, hence will be different for different architectures.

4.8 Improving the accuracy of the analytical model

It can be observed that the analytical model is developed based on a two-level approach. First, the channel distributions are considered to evaluate the trunk distributions at nodes and the mapping probabilities. Employing these trunk distribution and mapping probabilities, blocking performance on a path is obtained. The computation of blocking performance on a path does not explicitly include the channel distribution on the link. Hence, the analytical model developed in this case is not an exact computation of the blocking performance. However, we show that we obtain sufficient accuracy in estimating the path and tree blocking performance.

Some of the interesting features that are exhibited by networks cannot be observed if the analytical model is evaluated only once. For example, if a network rejects larger number of calls that travel longer distances as compared to another network, then it would accept larger number of calls that travel shorter distances. In order to obtain the finer behavior, the analytical model needs to be evaluated more than once by adjusting the parameters based on the results obtained in the earlier runs.

The two main input parameters of the analytical model are the link load and link load correlation. The computation of these two parameters are described in detail in Section 4.2. When calls are rejected by network, these two parameters are affected. The link load and link load correlation experienced by the network are only due to the calls that are accepted in the network. Hence, blocked calls do not have any effect on these parameters.

Let $P_{b,z}$ denote the probability of a call that requires a connection of length z hops and bandwidth of b channels. The adjusted link load seen in the network is computed as:

$$\lambda' = \frac{N\lambda_n \sum_{z=1}^{N-1} \sum_{b=1}^{B} zbp_{z,b}[1 - P_{b,z}]}{L} \tag{4.59}$$

The average path length and average capacity requirement of a call are adjusted as:

$$Z'_{av} = \sum_{z=1}^{N-1} \sum_{b=1}^{B} z\, p_{z,b}\, [1 - P_{z,b}] \tag{4.60}$$

$$B'_{av} = \sum_{z=1}^{N-1} \sum_{b=1}^{B} b\, p_{z,b}\, [1 - P_{z,b}] \tag{4.61}$$

The average resource required by a call is computed as:

$$R'_{av} = \sum_{z=1}^{N-1} \sum_{b=1}^{B} zbp_{z,b} \left[1 - P_{z,b}\right] \tag{4.62}$$

The adjusted link load correlation is obtained as:

$$\gamma'_c = \left(1 - \frac{B'_{av}}{R'_{av}}\right) \frac{1}{E} \tag{4.63}$$

where E denotes the number of exit links in the network.

The adjusted values for link load and link load correlation can be employed to evaluate the blocking performance iteratively. Such an iterative procedure could result in significant insights into the working of the network when the blocking probabilities are beyond a certain threshold. However, if the blocking probability values are very low, then the reduction in the link load and correlation values are not significant. Hence, such iterative procedures do not improve the accuracy of the model at low loads.

4.9 Summary

In this chapter, we developed an analytical model for evaluating the blocking performance in WDM grooming networks. The input to the analytical model are the link load and link load correlation. It is assumed that a fixed-path routing strategy is employed in the network and request arrival follows Poisson process. These assumptions are made in order to keep the analytical model tractable. Procedures to obtain link load and link load correlation values from request arrival rates at nodes are described. The analytical model considers: (1) heterogeneous architectures at nodes; (2) multiple arrival rates at nodes; and (2) multicast tree establishment. A methodology for iterative computation of blocking probability by adjusting the link load and correlation is also developed.

CHAPTER 5 Performance Evaluation

In this chapter, the accuracy of the analytical model is verified by comparing with simulations on different network architectures. Section 5.1 lists the properties of the different networks considered for performance evaluation. The simulation setup is described in Section 5.2. The blocking performance results obtained using analytical model and simulation are compared in Section5.3 under different categories. Section 5.4 outlines an iterative methodology that could be employed to improve the accuracy of the proposed model.

5.1 Networks considered and their parameters

In order to evaluate the accuracy of the proposed analytical framework, we consider a set of network topolgoies that constitute a good sample. The different network topologies that are considered are listed below with path length distribution and the number of exit links.

- **Uni-directional ring network with N nodes (N odd)**

$$P(z) = \begin{cases} \frac{1}{N-1} & 1 \leq z \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

$$E = 1 \tag{5.2}$$

- **Bi-directional ring network with N nodes (N odd)**

$$P(z) = \begin{cases} \frac{2}{N-1} & 1 \leq z \leq \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

$$E = 1 \tag{5.4}$$

- **Bi-directional $M \times M$ mesh-torus network (M odd)**

$$P(z) = \begin{cases} \frac{4z}{M^2-1} & \text{if } 1 \leq z \leq \frac{M-1}{2} \\ \frac{4(M-z)}{M^2-1} & \frac{M-1}{2} < z \leq M-1 \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

$$E = 3 \quad (5.6)$$

- **Uni-directional $R \times C$ mesh-torus network**

$$P(z) = \begin{cases} \frac{z+1}{RC-1} & 1 \leq z < \min(R,C) \\ \frac{\min(R,C)}{RC-1} & \min(R,C) \leq z < \max(R,C) \\ \frac{R+C-z-1}{RC-1} & \max(R,C) \leq z < R+C-1 \\ 0 & \text{otherwise} \end{cases} \quad (5.7)$$

$$E = 2 \quad (5.8)$$

The above choice of networks are motivated based on the average length of shortest path between node pairs, hence the link load correlation. Ring networks have longer path lengths, hence have higher link load correlation. Mesh-torus with the same number of nodes as a ring network has smaller average shortest path length due to the increased connectivity, hence has low values of correlation. Table 5.1 lists the specific network that are considered in the above mentioned network topologies and their respective average length of shortest path and link load correlation.

5.2 Simulation setup

A network simulation framework has been developed in order to evaluate the performance of different routing algorithms on various switching architectures and network topologies. A network is comprised of Source/Destination nodes that can generate requests and act as destination for requests generated at other Source/Destination nodes. A network also comprises of dummy nodes that switch traffic across and are not capable of generating requests or being destination of requests generated at other nodes. With the use of the above two kinds of nodes it has been shown that nodes

Table 5.1 Networks with their average shortest path length and link load correlation.

Network	Average shortest path length	Correlation
25-node bi-directional ring	6.5	0.8462
25-node uni-directional ring	12.5	0.92
11-node uni-directional ring	5.5	0.8182
5×5 bi-directional mesh-torus	2.5	0.2
7×7 bi-directional mesh-torus	3.5	0.2381
3×3 unidirectional mesh-torus	2.25	0.2778
3×5 uni-directional mesh-torus	3.214	0.3444
3×6 uni-directional mesh-torus	3.706	0.3651

channel-space switch can be modeled. Such a network specification allows to model networks with heterogeneous switching architecture.

The requests are generated according to a Poisson process with a rate $N_s\lambda$, where N_s denotes the number of Source/Destination nodes. Any Source/Destination node in the network is equally likely to be the source of the request. The destination of a request is uniformly selected from the remaining Source/Destination nodes. A request that is generated is fed to multiple networks that are derived from a physical network but with differing network architectures. The requests are considered by different networks independently. The networks are assumed to employ shortest-path routing strategy. If more than one path with the minimum path length is available, then one of them is selected at random. A connection is attempted along the path. If sufficient resources are available, then the call is accepted. Otherwise, the call is rejected. It is to be noted that not all the paths with the minimum length are attempted for establishing the call. It is possible that among the paths that have the minimum length, there is one path that could accommodate the call but is not chosen when selected randomly.

The experiments are run for a total of 500,000 requests with performance metrics obtained after every 100,000 requests resulting in five sets of values. The performance results shown in the rest of this chapter is the average of the five sets of results. The blocking probability is obtained as the fraction of the number of call blocked to the total number of calls received in the network.

We observe the blocking probability of calls that travel a specific hop length. This observation can be made in two ways. First, requests can be specifically classified into groups based on the shortest path length between the source and destination and blocking probability can be computed on individual groups. While this is a good mechanism to certain specific events, it is not a practical solution for observing rare events. For example, assume that blocking performance from a source to destination on a specific path from a source to destination needs to be computed.

Another approach to measure the blocking performance on a path is to monitor the path on every event occurrence in the network. The events in the network are either of a new request or termination of an already existing connection. Before accepting a new connection or terminating an already existing connection, the specific path is checked for availability. The network has remained in this state starting from the previous event occurrence until the current time. If the path is available, then the difference in time between the current time and the previous event occurrence time is accumulated in the path available time, otherwise it is accumulated to the path unavailable time. The ratio of the path unavailable time to the sum of the path available and unavailable times gives the blocking probability.

We employ the former approach to obtain blocking probabilities for paths of certain length. We employ the latter approach to obtain blocking probabilities for establishing trees as muticast connections are rare events as compared to unicast connections. We also employ the latter approach to obtain blocking probability of a specific path in a heterogeneous network.

5.3 Performance results

We compare the performance results obtained from analytical model and simulations under different sections, each dealing with specific aspect of the analytical model. We compare the blocking probabilities of calls that need to travel over a specific hop length.

5.3.1 Homogeneous networks

We consider two networks: (1) 25-node bi-directional ring network; and (2) 5×5 bi-directional mesh-torus network. Three different trunk-channel combinations are considered: (1) 1 trunk with 20 channels; (2) 2 trunks with 10 channels each; and (3) 4 trunks with 5 channels each. Two different switch architectures are considered: (1) full-permutation switching per trunk (FP); and (2) channel-space switching (CS).

We first consider full-permutation switching per trunk employed at every node in the network. Figures 5.1 through 5.4 show the blocking performance with respect to different path lengths obtained through analytical model and simulation for total network load of 60, 80, 100, and 120 Erlangs, respectively. The simulation results are shown in points, while the lines show the values predicted by the analytical model. It is observed that the accuracy of the analytical model improves with the increase in path length. For calls with path length of one hop, the analytical model predicts blocking probability that is approximately a factor of 4 higher than the simulation values. It is also observed that the accuracy of the analytical model improves with the decrease in the number of trunks.

One of the effects that is observed with the different trunk and channel combinations is that a network that blocks more calls of longer path lengths tends to accept more calls of shorter path lengths. These are revealed in simulation, however the trends are not reflected in the analytical model. If the nodes in a network view a link as more trunks with fewer channels in each, then it reflects a lower switching capability in the network. In such networks, calls that require longer paths will experience more blocking as compared to a network that has more switching capability. Because of this, a network with lower switching capability would accept more calls traveling over a fewer number of hops as compared to networks with higher switching capability. Analytical models do not predict such a trend when they are employed once. Analytical models directly predict the capability of the switch for a specific load in the network. In order to obtain the trend as seen in the simulation, the link load and the correlation has to be computed analytically based on the blocking probabilities obtained. The computed average link load and the correlation can

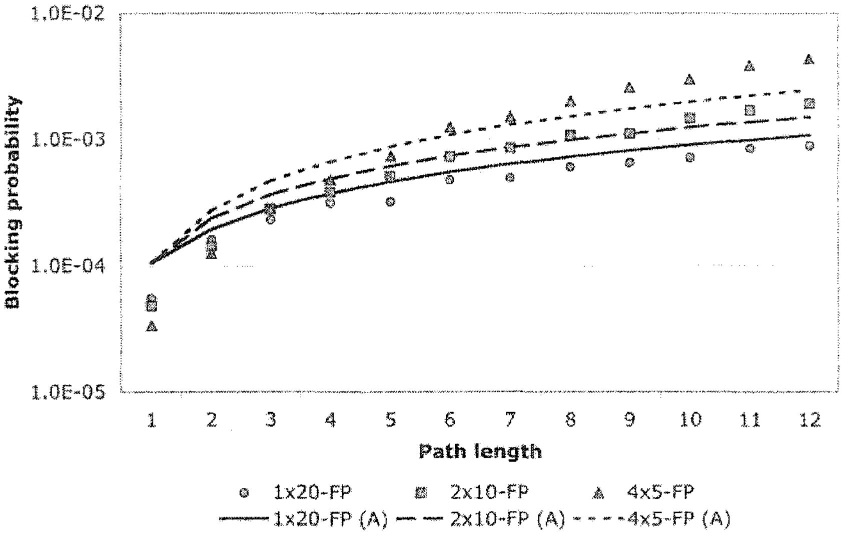


Figure 5.1 Blocking performance of 25-node bi-directional ring network with full permutation switching per trunk for calls of varying path lengths for a network load of 60 Erlangs (link load of 7.8 Erlangs).

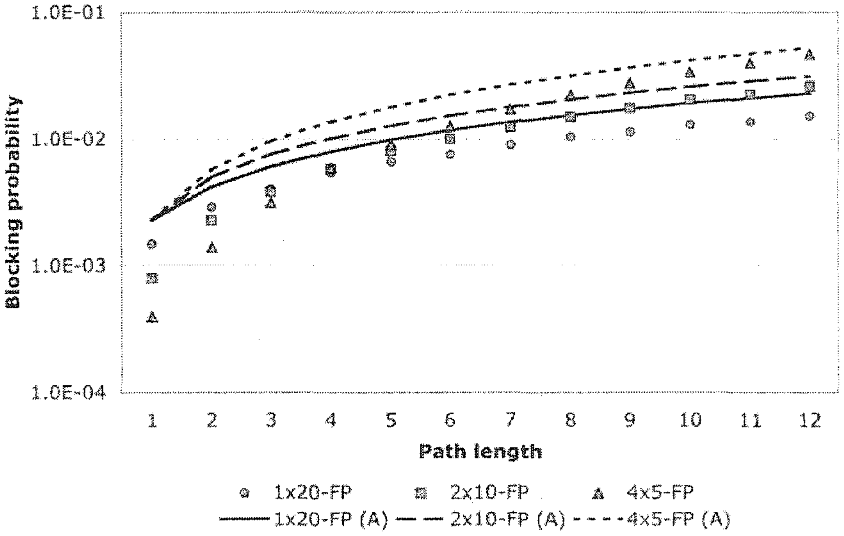


Figure 5.2 Blocking performance of 25-node bi-directional ring network with full permutation switching per trunk for calls of varying path lengths for a network load of 80 Erlangs (link load of 10.4 Erlangs).

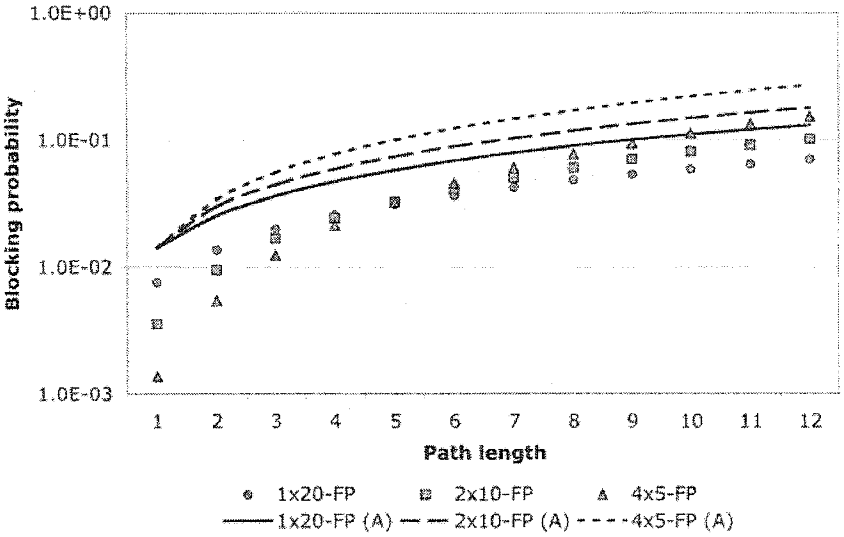


Figure 5.3 Blocking performance of 25-node bi-directional ring network with full permutation switching per trunk for calls of varying path lengths for a network load of 100 Erlangs (link load of 13 Erlangs).

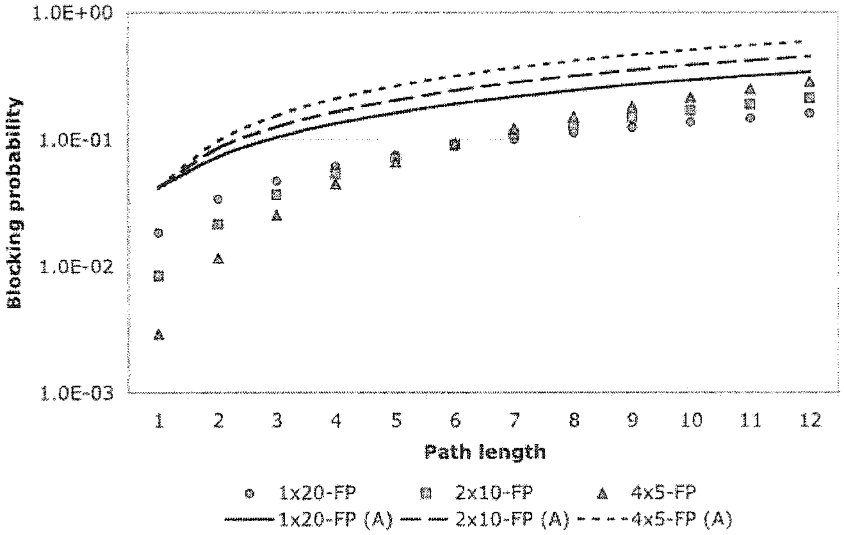


Figure 5.4 Blocking performance of 25-node bi-directional ring network with full permutation switching per trunk for calls of varying path lengths for a network load of 120 Erlangs (link load of 15.6 Erlangs).

be used to recompute the blocking probabilities. Improving the accuracy of the analytical model based on such an iterative procedure involving computation of link loads and link correlation is discussed in detail in Section 5.4.

Figures 5.5 through 5.8 shows the blocking performance on a 5×5 bi-directional mesh-torus network for calls of different path lengths for offered network loads of 400, 450, 500, and 550 Erlangs (corresponding to link loads of 10, 11.25, 12.5, and 13.75 Erlangs), respectively. It is observed that the analytical model closely matches the simulation results. As observed in the ring network, the analytical model predicts the same blocking performance for different trunk and channel combinations for calls that travel shorter lengths.

We consider the second switching architecture, namely the channel-space switch. In order to simulate a channel space switch, we model every node in the physical network as one Source/Destination node with a few dummy nodes as described in Section 3.4. It is to be noted that despite the full channel interchanger at the input of every trunk, channel continuity has to be satisfied at the switch. Hence, these switches are expected to block more calls as compared to the full-permutation switches.

Figure 5.9 through 5.12 show the blocking performance of a 25-node bi-directional ring network with channel space switch for calls of different path lengths. It is again observed that the analytical model predicts well at low loads and the accuracy drops as the load increases.

Comparing the performance of the channel-space switch with that of the full-permutation switching (refer to Figures 5.1 through 5.4 for performance of full-permutation switching), it is observed that the blocking performance when employing channel-space do not vary significantly. Hence, in networks with higher link load correlation, channel-space switches could be employed with minimal impact in blocking performance. Note that, a channel-space switch has a lower implementation complexity as compared to a full-permutation switch. The increase in the complexity of the switches would necessitate power compensation and clock synchronization at different stages in the switch, thus increasing the cost of the switch.

We also evaluate the blocking performance on a 3×3 uni-directional mesh-torus network. The

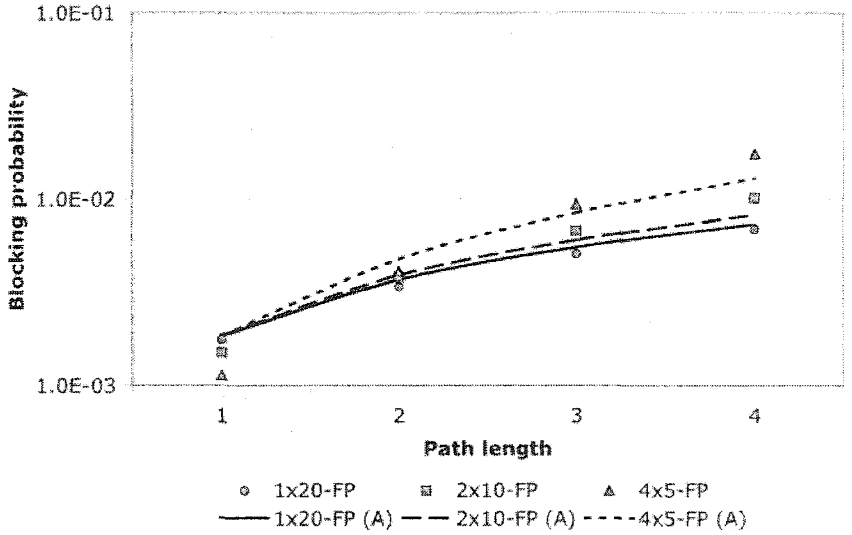


Figure 5.5 Blocking performance of 5×5 bi-directional mesh-torus network for calls of varying path lengths for a network load of 400 Erlangs (link load of 10 Erlangs).

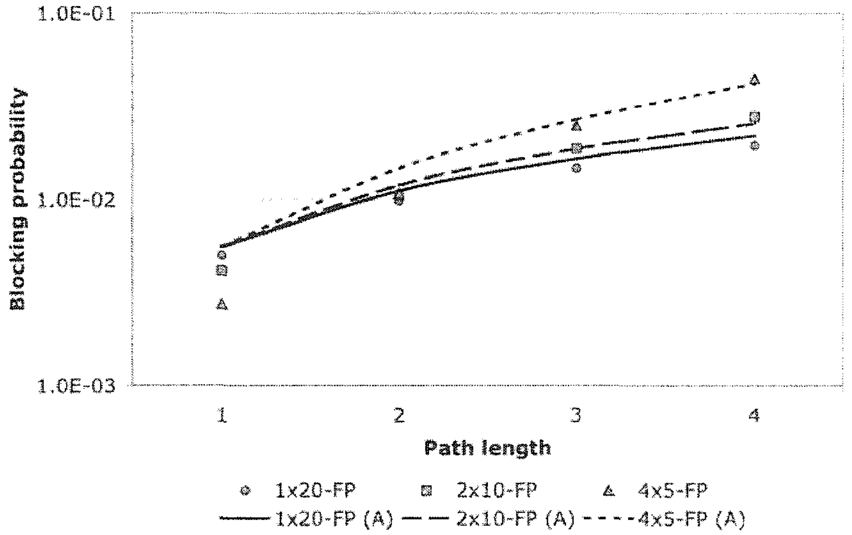


Figure 5.6 Blocking performance of 5×5 bi-directional mesh-torus network for calls of varying path lengths for a network load of 450 Erlangs (link load of 11.25 Erlangs).

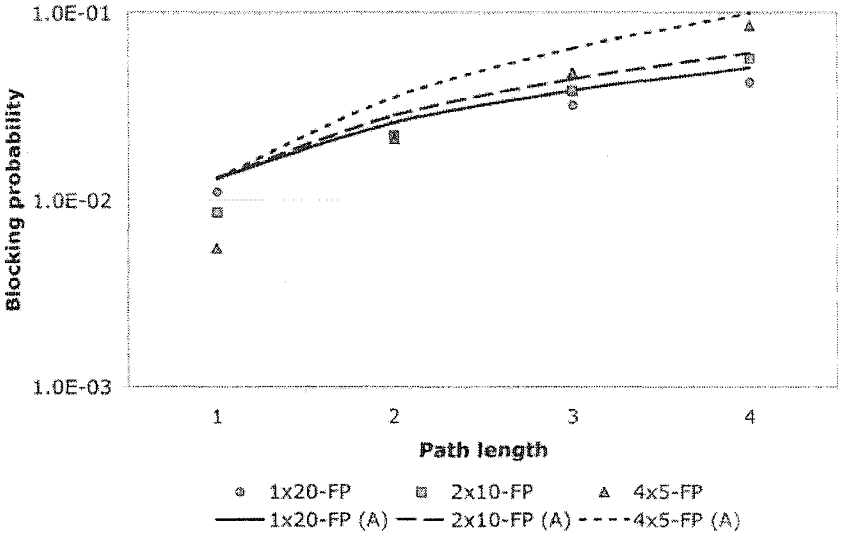


Figure 5.7 Blocking performance of 5×5 bi-directional mesh-torus network for calls of varying path lengths for a network load of 500 Erlangs (link load of 12.5 Erlangs).

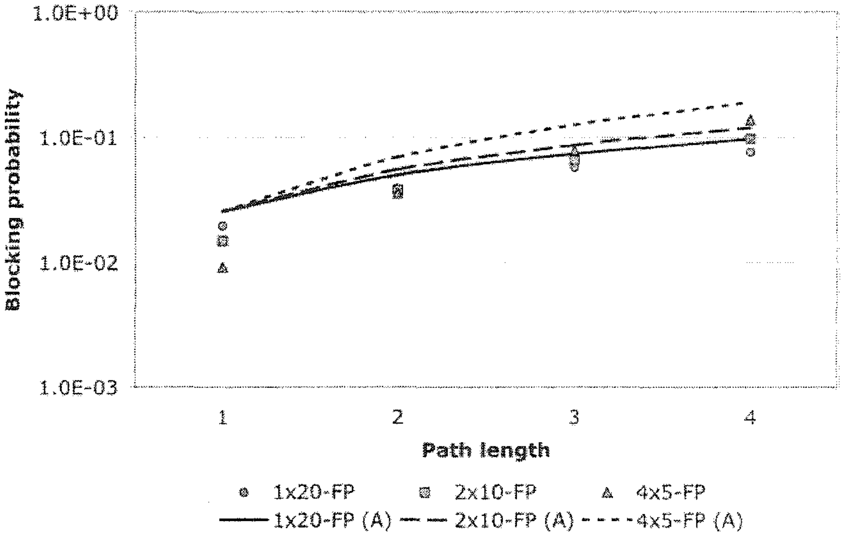


Figure 5.8 Blocking performance of 5×5 bi-directional mesh-torus network for calls of varying path lengths for a network load of 550 Erlangs (link load of 13.75 Erlangs).

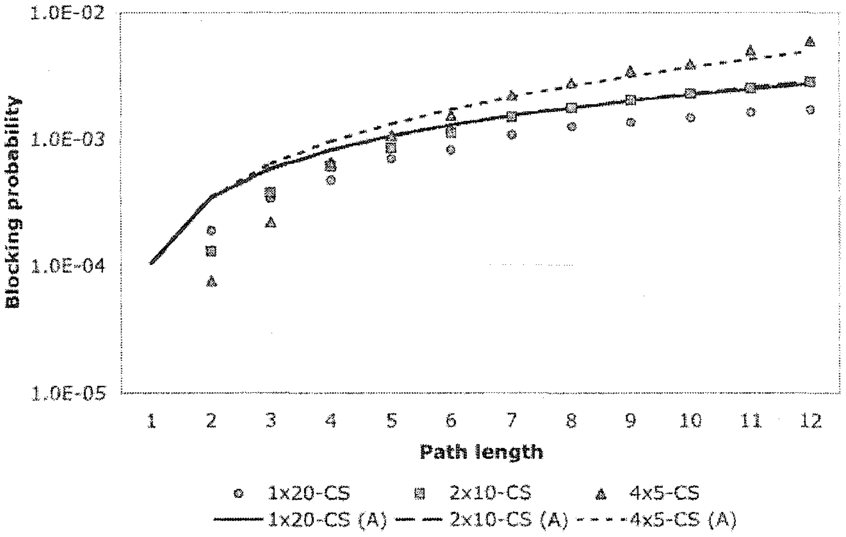


Figure 5.9 Blocking performance of 25-node bi-directional ring network, employing channel-space switching, for calls of varying path lengths for a network load of 60 Erlangs (link load of 7.8 Erlangs).

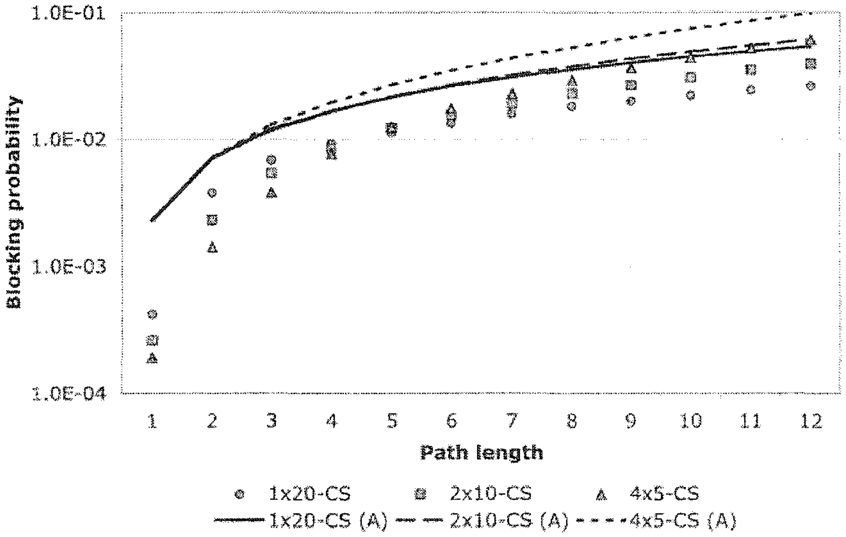


Figure 5.10 Blocking performance of 25-node bi-directional ring network, employing channel-space switching, for calls of varying path lengths for a network load of 80 Erlangs (link load of 10.4 Erlangs).

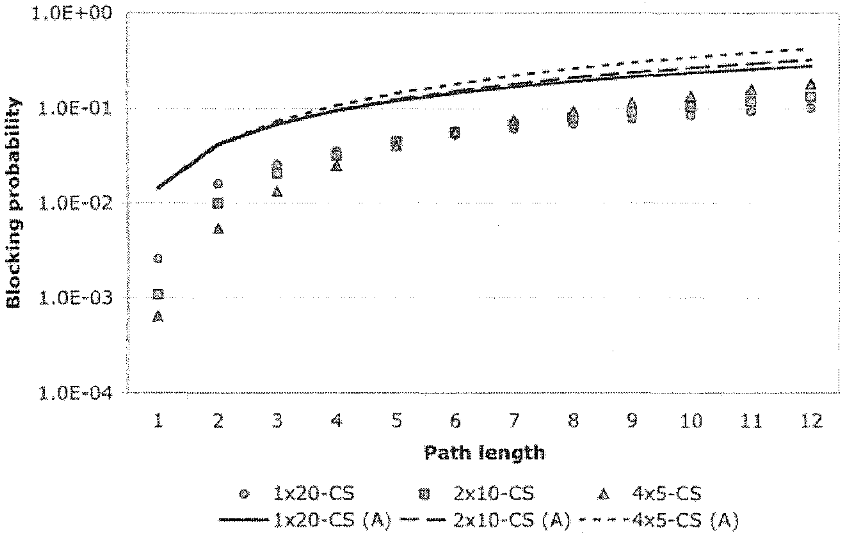


Figure 5.11 Blocking performance of 25-node bi-directional ring network, employing channel-space switching, for calls of varying path lengths for a network load of 100 Erlangs (link load of 13 Erlangs).

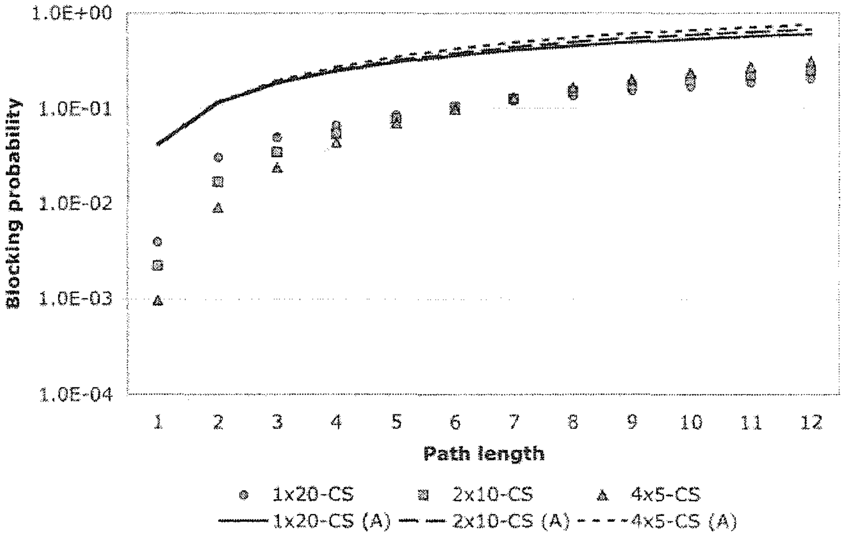


Figure 5.12 Blocking performance of 25-node bi-directional ring network, employing channel-space switching, for calls of varying path lengths for a network load of 120 Erlangs (link load of 15.6 Erlangs).

choice of this network is due to its similarity to a 5×5 bi-directional mesh-torus network. Both the networks have the same maximum hop-length. The distribution of the hop-lengths are different, hence the link load correlation for the two networks differ, but not significantly. As the channel-space switch has limited switching capability, the blocking probability increases at a higher rate with increasing path length as compared to a full-permutation switch. Hence, as the correlation in the network decreases, the individual link loads become more independent, hence the blocking on longer paths increase. The choice for these two networks are due to simulation constraints. Note that every node in the network connected to a link is replaced with a dummy node an an extra link. Hence, a 5×5 bi-directional mesh-torus network would be modeled as 25 Source/Destination nodes with each node having 4 dummy nodes and four extra links. This implies that the network will have in total 125 nodes with 400 links with each link being represented as matrix. Due to heavy computational requirements, these simulations were unable to be run. Hence, to show the effect on network with lower correlation, a 3×3 uni-directional mesh-torus network.

Figures 5.13 through 5.16 shows the blocking performance of a 3×3 uni-directional mesh-torus network with nodes employing channel-space switching for offered network loads of 9, 9.75, 10.5, and 11.25 Erlangs, respectively. From the figures it is observed that the analytical model closely approximates the simulation results. For connections with path length of one hop, the blocking probability predicted by the analytical model is approximately a factor of 4 higher than the simulation values. This is again due to the effect of networks with lower switching capability having a tendency to accept more connections of shorter path lengths as compared to networks with higher switching capability. From path lengths of two through four, the analytical model gives a good estimate of the blocking performance.

For network with lower link load correlation, the rate of increase of blocking performance with the increase in path length is higher. One of trends of blocking observed with the channel-space switch is that the blocking performance varies up to two orders of magnitude between path lengths of one and four. Comparing this performance with the performance of full-permutation switching in a 5×5 bi-directional mesh-torus network (Figures 5.5 through 5.8, the rate of increase

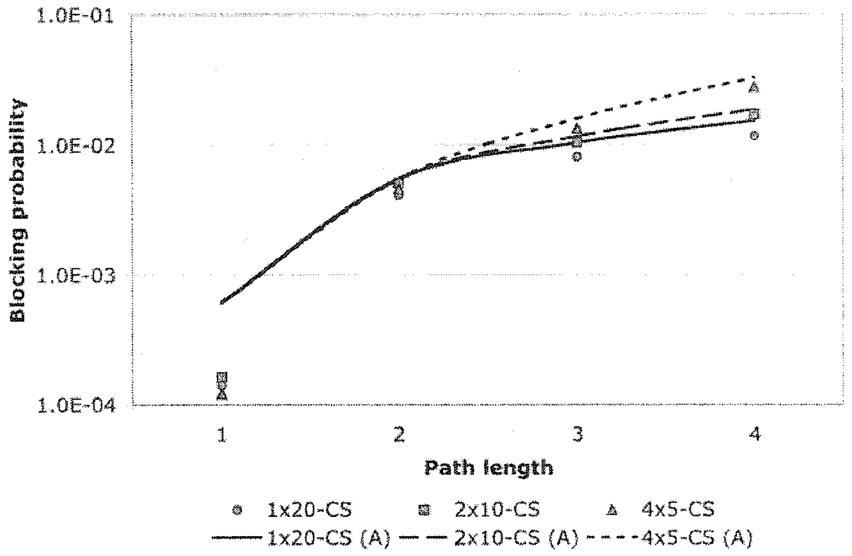


Figure 5.13 Blocking performance of 3×3 mesh-torus network, employing channel-space switching, for calls of varying path lengths for a network load of 72 Erlangs (link load of 9 Erlangs).

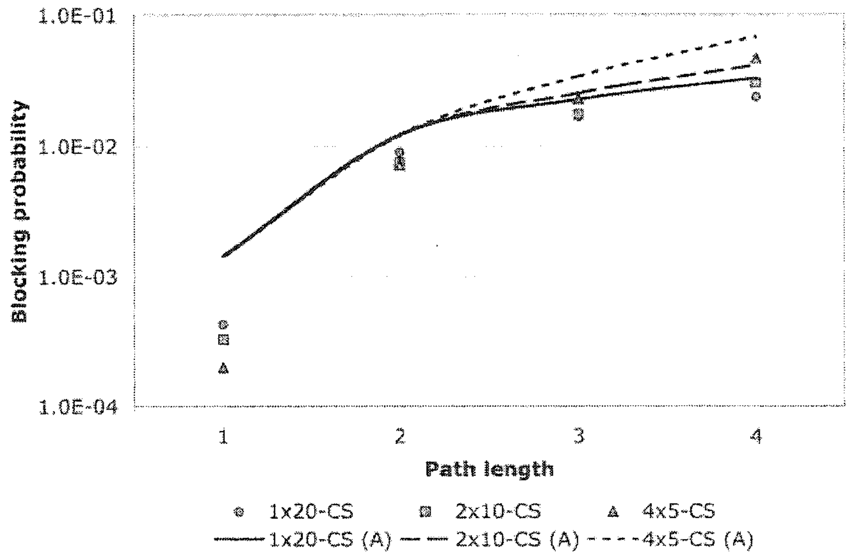


Figure 5.14 Blocking performance of 3×3 mesh-torus network, employing channel-space switching, for calls of varying path lengths for a network load of 78 Erlangs (link load of 9.75 Erlangs).

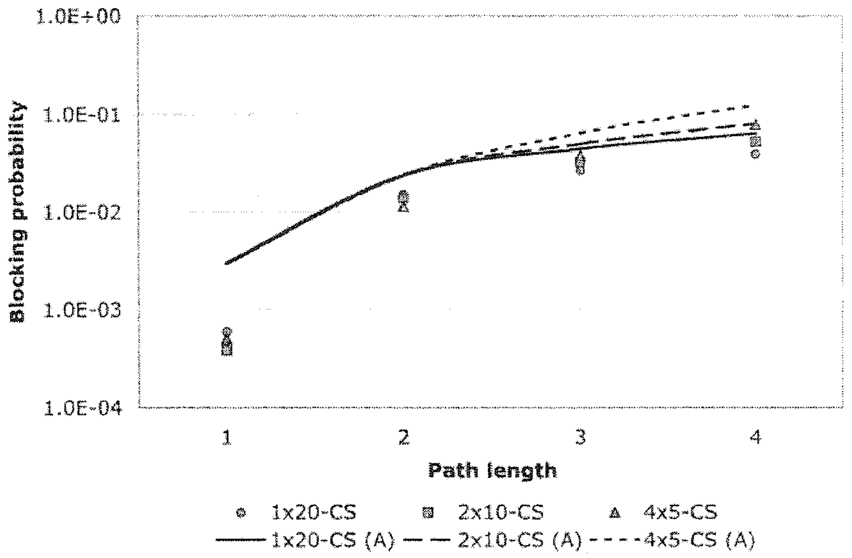


Figure 5.15 Blocking performance of 3×3 mesh-torus network, employing channel-space switching, for calls of varying path lengths for a network load of 84 Erlangs (link load of 10.5 Erlangs).

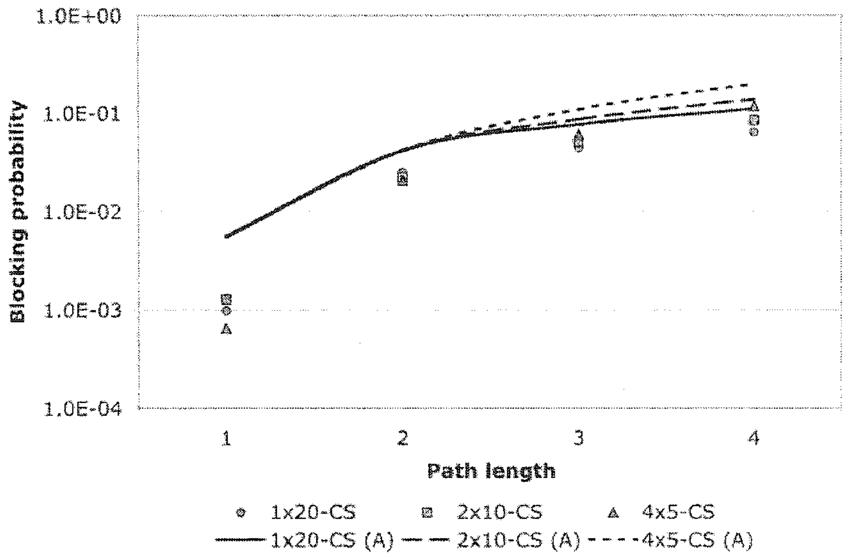


Figure 5.16 Blocking performance of 3×3 mesh-torus network, employing channel-space switching, for calls of varying path lengths for a network load of 90 Erlangs (link load of 11.25 Erlangs).

due to the full-permutation switch is lower. Also, note that the link load correlation of a 5×5 bi-directional mesh-torus network is lower than that of 3×3 uni-directional mesh-torus network. While this observation favors employing full-permutation switching, channel-space switching with other trunk assignment algorithms such as first-fit, best-fit, etc. or re-arranging the connections would help reduce the blocking performance even when channel-space switching is employed.

An important observation that could be made from the performance of homogeneous networks is that the blocking performance of establishing unicast connections under different grooming capabilities do not vary significantly. The difference in blocking performance is well within an order of magnitude. Therefore, in networks that have predominantly unicast connections requiring one time slot capacity, improving the grooming capability of the network may not result in a significant improvement in blocking performance. For example, from the performance graphs presented in this chapter thus far, it can be observed that significant gain in performance is not achieved by employing one wavelength with twenty time slots (1×20) as compared to four wavelengths with five time slots (4×5). The advantage of employing four wavelengths is that the switching speed in the network can be four times as slow as that employed in a single-wavelength network.

5.3.2 Tree establishment in homogeneous networks

We evaluate the blocking performance of establishing multicast trees on 5×5 and 7×7 bi-directional mesh-torus networks with nodes employing full permutation switching for every trunk. We study the blocking performance of establishing trees with 4, 6, 8, and 10 links with the tree branching randomly after the first link. We observe a set of links in the simulation to check if the tree can be established at every event occurrence in the network.

Figures 5.17 through 5.20 show the blocking performance of establishing trees with 4, 6, 8, and 10 links, respectively, on a 5×5 bi-directional network. Figures 5.21 through 5.24 show the blocking performance of establishing trees with 4, 6, 8, and 10 links, respectively, on a 7×7 bi-directional network.

The figures show the simulation results for trees of with varying number links and the analyt-

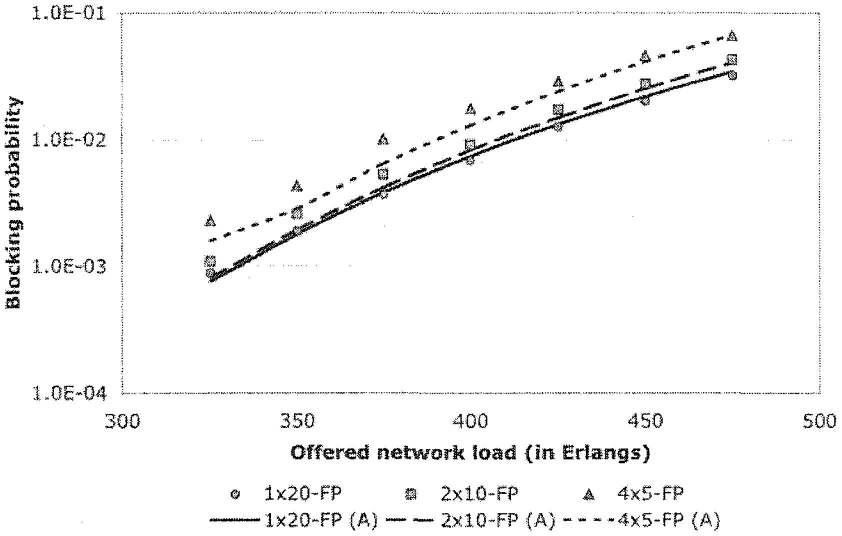


Figure 5.17 Blocking performance for establishing a tree with 4 links in a 5×5 homogeneous bi-directional mesh-torus network.

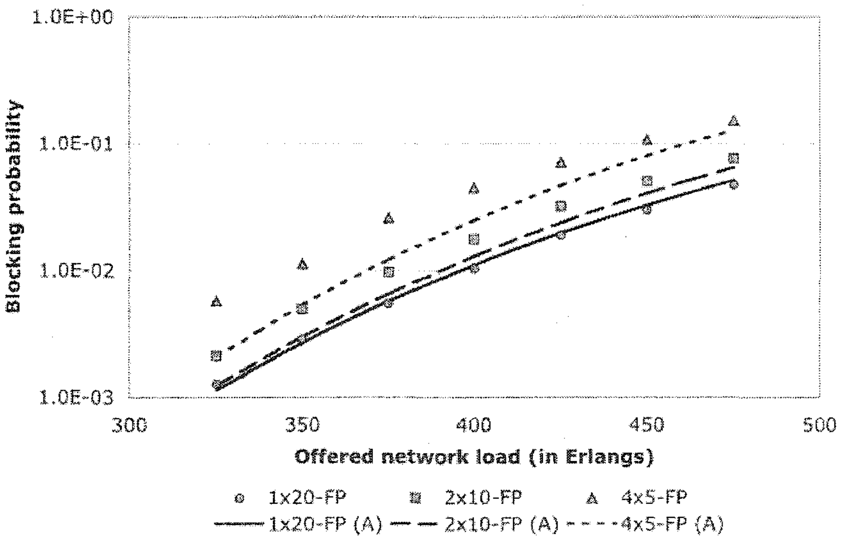


Figure 5.18 Blocking performance for establishing a tree with 6 links in a 5×5 homogeneous bi-directional mesh-torus network.

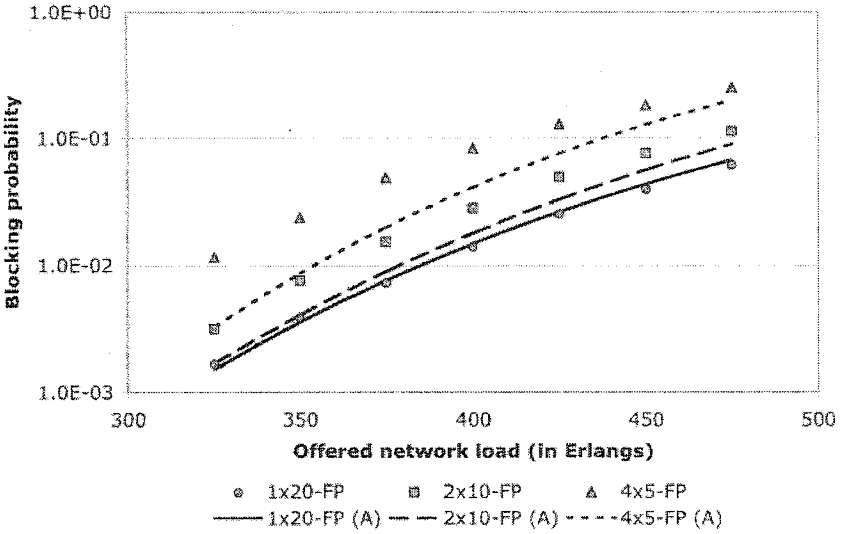


Figure 5.19 Blocking performance for establishing a tree with 8 links in a 5×5 homogeneous bi-directional mesh-torus network.

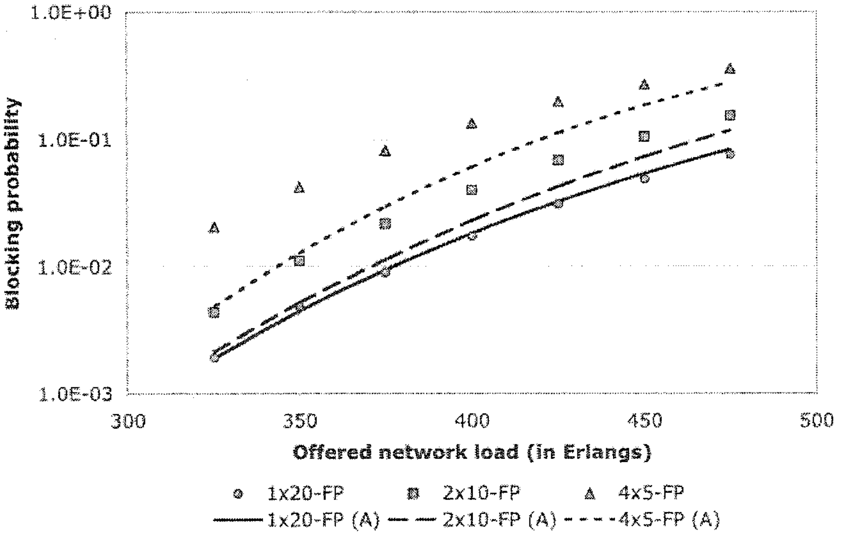


Figure 5.20 Blocking performance for establishing a tree with 10 links in a 5×5 homogeneous bi-directional mesh-torus network.

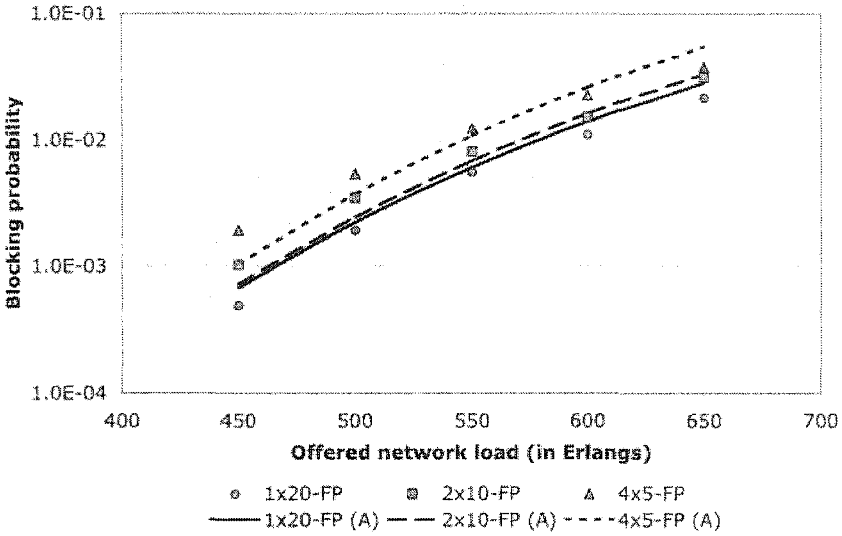


Figure 5.21 Blocking performance for establishing a tree with 4 links in a 7×7 homogeneous bi-directional mesh-torus network.

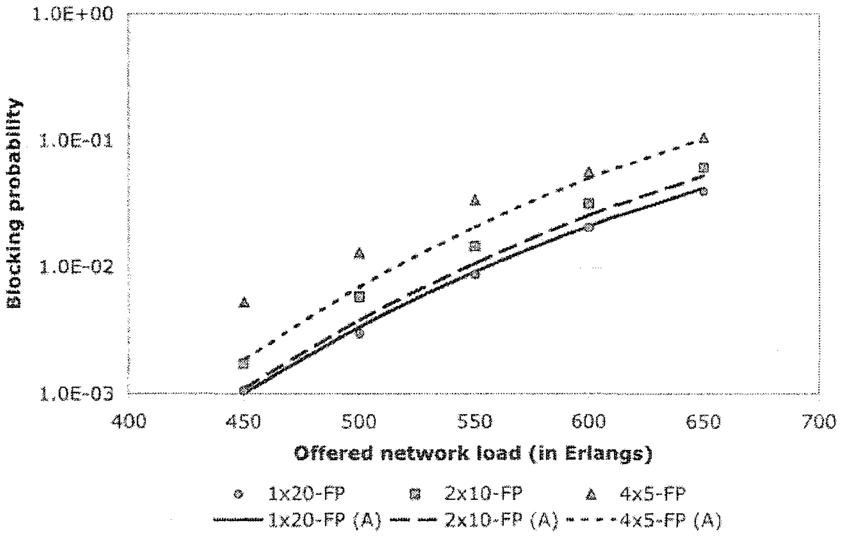


Figure 5.22 Blocking performance for establishing a tree with 6 links in a 7×7 homogeneous bi-directional mesh-torus network.

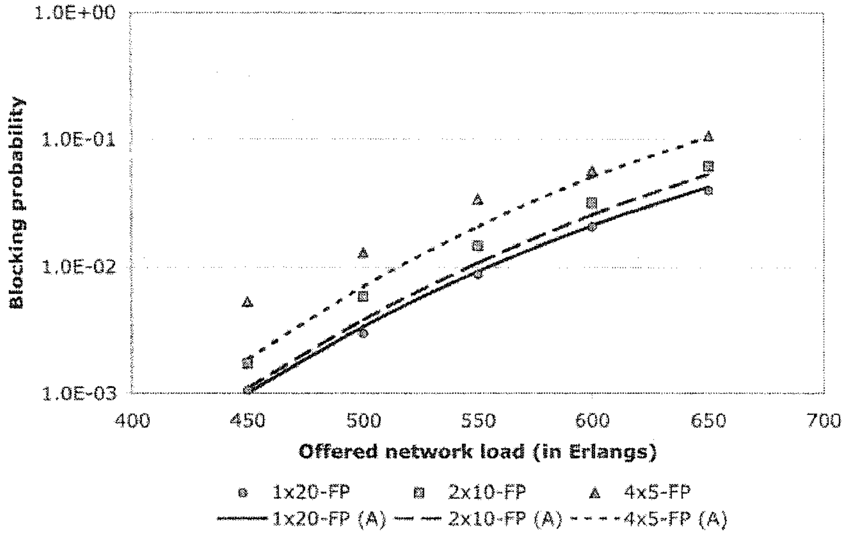


Figure 5.23 Blocking performance for establishing a tree with 8 links in a 7×7 homogeneous bi-directional mesh-torus network.

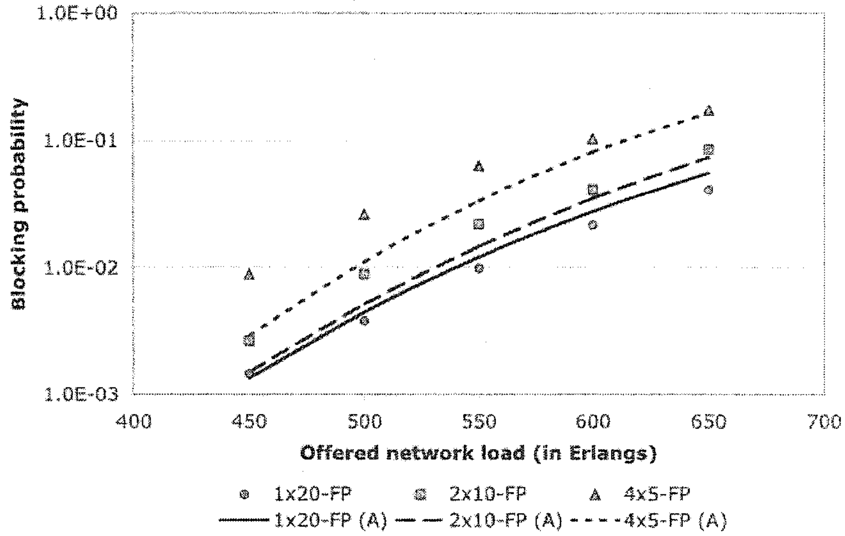


Figure 5.24 Blocking performance for establishing a tree with 10 links in a 7×7 homogeneous bi-directional mesh-torus network.

ical model values are obtained through the path model. It is observed the blocking performance predicted by the analytical model for path and tree are approximately the same. Hence, they are not plotted in the above mentioned figures. It is also observed through simulations that the a tree and a path with the same number of links have almost the same blocking probability of being established in a network. Again, as these values are too close to distinguish when plotted in graphs. Hence, simulation values for paths of varying lengths are not shown in the above mentioned figures.

It is to be noted that the maximum path length of unicast connections when established under shortest path routing algorithm on 5×5 and 7×7 bi-directional mesh-torus networks are 4 and 6, respectively. It is observed that the analytical model has good accuracy in predicting the blocking performance of paths that are beyond the maximum path length of unicast connections.

It is observed from the performance results that the blocking probability when employing one wavelength and twenty time slots could vary up to an order of magnitude. However, it is to be noted that multicast connections form a very small fraction of the network traffic. Hence, mechanisms such as allocation of resources specific to multicast connections could be employed to improve network performance. Multicast connections also require slot copying feature that would be difficult and expensive to implement. Hence, it may not be possible to employ such a feature for all the wavelengths (or trunks) in the networks. The expected high cost of slot copying feature also indicates the need to investigate into mechanisms that would dedicate a fraction of the network resources to multicast connections.

In networks that have a very high link correlation, it is intuitive that a tree can be approximated to a path as there will be lesser branching. This is not intuitive for networks with lower correlation, such as mesh-torus networks. With the analytical model developed in the simulation, it is shown that the analytical model for evaluating path blocking performance can approximate that developed for evaluating tree blocking performance.

5.3.3 Dual-rate traffic in homogeneous networks

We consider establishing calls that are classified into two categories, namely calls that require one time slot and those that require two time slots. We evaluate the blocking performance of establishing dual rate calls in 25-node homogeneous bi-directional ring network and 5×5 homogeneous bi-directional mesh-torus network employing full-permutation switching per trunk.

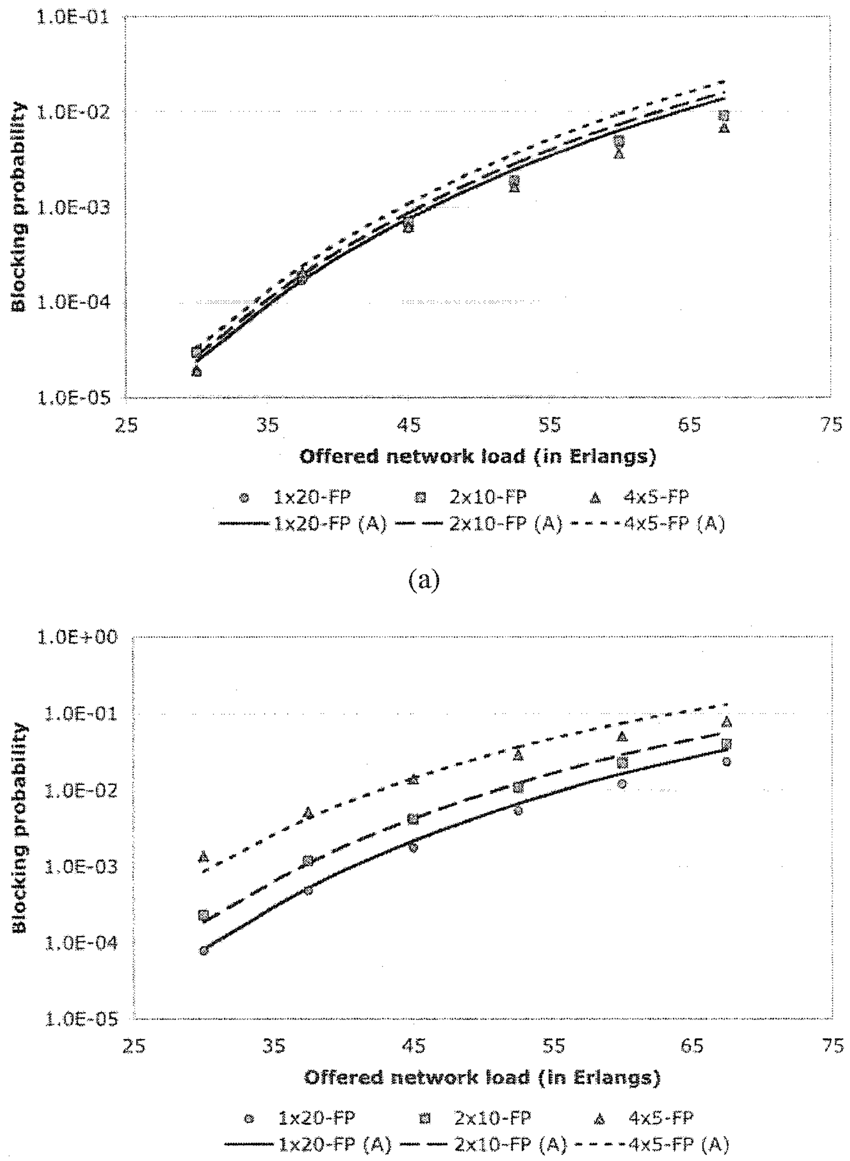
It is assumed that every call request in the network has equal probability of requesting for one or two time slots. It is also assumed that the two time slots required by a connection must be provided in the same trunk. Splitting of a call requesting for two time slots into two calls of one time slot requirement is not permitted.

Figures 5.25 and 5.26 show the comparison of blocking performance obtained through simulation and analysis for 25-node homogeneous bi-directional ring network and 5×5 homogeneous bi-directional mesh-torus network, respectively.

It is observed that the analytical model closely approximates the simulation values. It is also observed that for the blocking performance for calls with one time slot requirement, networks with lower switching capability perform better. This is due to the reason that these networks block more calls that require two time slots, thereby accept more calls requiring one time slot.

The performance obtained by increasing grooming capability is observed to be within an order of magnitude for dual-rate connections. However, if calls with higher capacity requirements are considered, then networks with lower grooming capability would perform significantly worse than those with higher grooming capability, especially for requests with large capacity requirement. In such networks, special methodologies such as *dispersity routing* can be adopted to improve the network performance with lower grooming capability at the switches. Such an approach is considered in detail in Chapter 6.

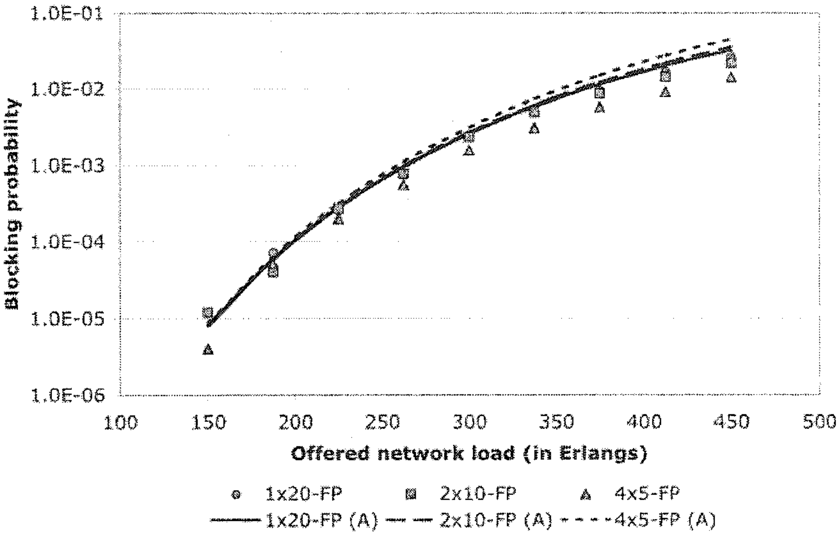
The analytical model developed in this dissertation assumes fixed-path routing strategy for the sake of tractability. Employing dynamic routing algorithms would improve network performance and is dealt with in detail in Chapter 6.



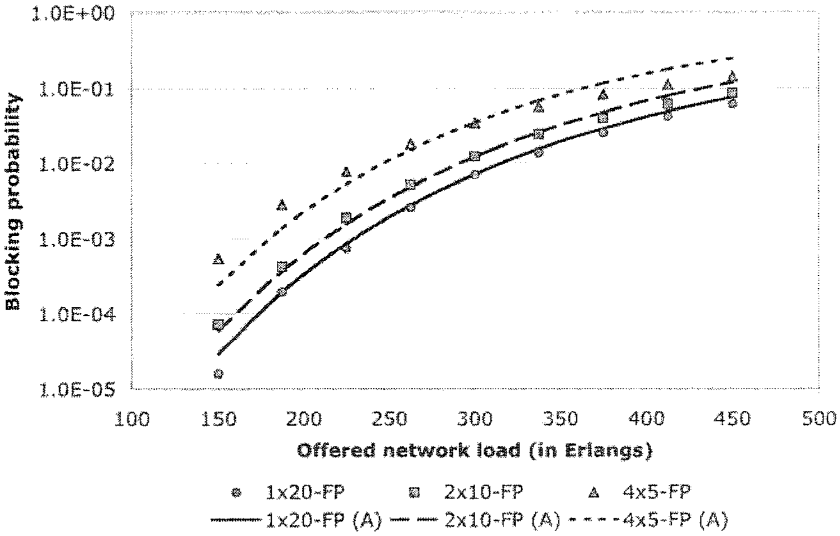
(a)

(b)

Figure 5.25 Blocking performance for establishing calls in a 25-node homogeneous bi-directional ring network that require (a) one time slot; and (b) two time slots.



(a)



(b)

Figure 5.26 Blocking performance for establishing calls in a 5×5 homogeneous bi-directional mesh-torus network that require (a) one time slot; and (b) two time slots.

5.3.4 Heterogeneous networks

We evaluate the blocking performance ring and mesh-torus network architectures with nodes employing heterogeneous switching and grooming architectures.

We study a 9-node uni-directional ring network and 3×3 uni-directional mesh-torus networks. We consider a link with 20 channels organized as two fibers, five wavelengths per fiber, and two time slots per wavelength. A node can classified into any one of the following categories based on the level of grooming: (1) time-slot level grooming node; (2) wavelength-level grooming node; and (3) full-grooming node. A time slot-level grooming node would view the link as two trunks with ten channels each. A wavelength level-grooming node views a link as five trunks with four channels each while a full grooming network views a link as one trunk. We consider two different switching architectures employed at a node for a trunk: (1) full-permutation; and (2) channel-space switch. In the experiments that we discuss here, we assume that all the nodes employ similar switching architecture within a trunk, while the trunk definition could vary.

The 9-node uni-directional ring network and 3×3 uni-direcitonal mesh-torus network with heterogeneous node architectures are organized as shown in Figures 5.27 with nodes of same architecture equally spaced.

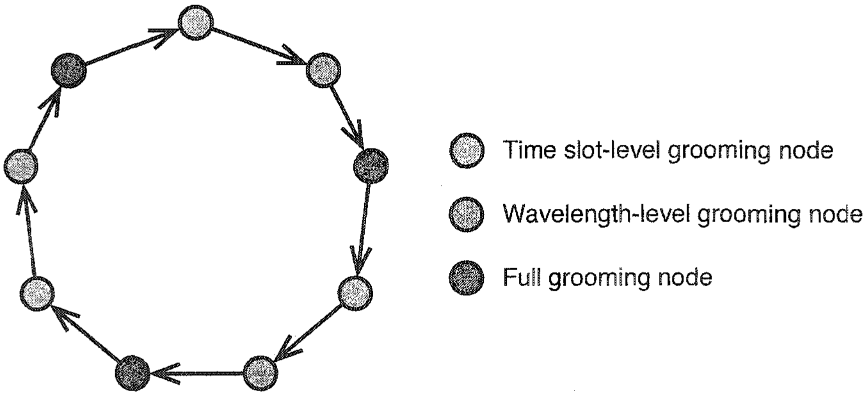


Figure 5.27 A 9-node heterogeneous uni-directional ring network with three different switch architectures.

It can be seen that any path with a certain path length can be classified into three categories

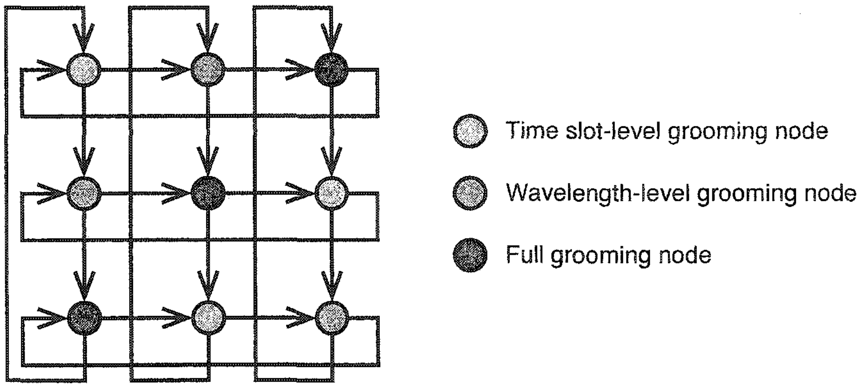


Figure 5.28 A 9-node heterogeneous bi-directional ring network with three different switch architectures.

depending on the source. We refer to a path originating from a time-slot level grooming node as Path-1, wavelength-level grooming node as Path-2, and full-grooming node as Path-3.

We assume that all calls have a requirement of one time slot capacity. The results of the analytical model to be shown in the graphs are obtained without employing the knowledge about the trunk distribution for mapping the trunk distribution between adjacent nodes. The difference in blocking performance obtained with and without using the exact trunk information is found to be less than 2%. Hence, these are not plotted in the graphs for the sake of clarity.

Figures 5.29 through 5.32 show the blocking performance of a 9-node uni-directional ring network with nodes employing full-permutation switching in each trunk for three different path types with varying path lengths for offered network load of 15, 18, 21, and 24 Erlangs (link loads of 7.5, 9, 10.5, and 12 Erlangs), respectively.

It is observed that the performance trend observed with the simulation for the different path lengths is also observed through the analysis. The blocking performance as estimated by the analysis is the same for all the paths with a length of one hop. This is due to the reason that a single hop blocking performance remains the same for a given link load and correlation factor for any switch architecture. For two-hop paths, the blocking performance depends on the switching capability of intermediate node. Calls that would have the intermediate node as a full grooming node (Path-

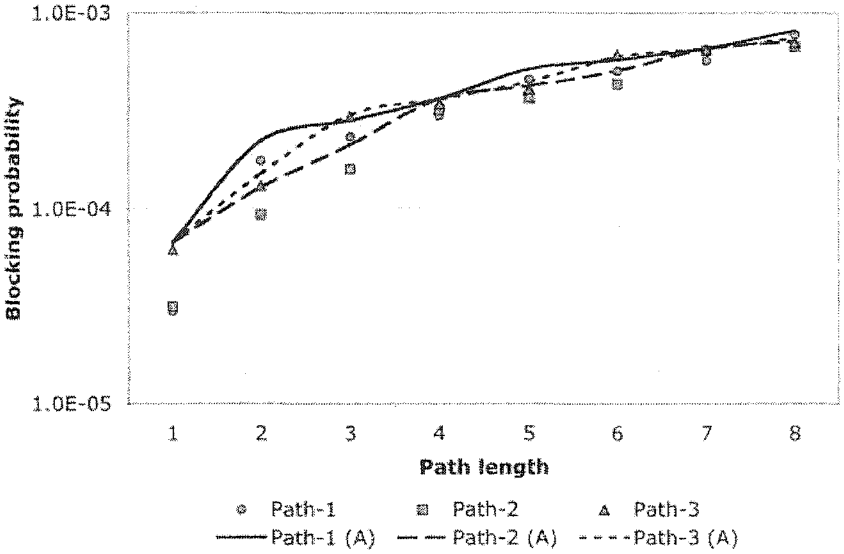


Figure 5.29 Blocking performance of 9-node heterogeneous uni-directional ring network with nodes employing full-permutation switching in each trunk for an offered network load of 15 Erlangs.

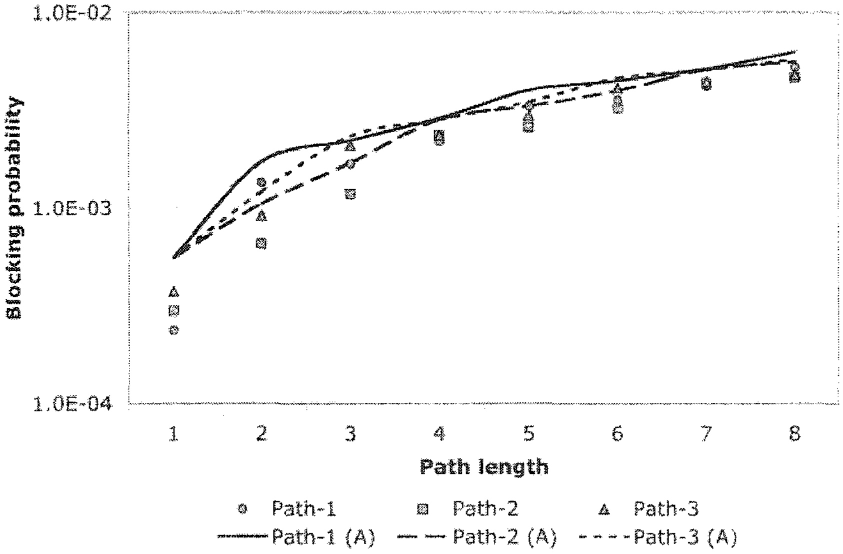


Figure 5.30 Blocking performance of 9-node heterogeneous uni-directional ring network with nodes employing full-permutation switching in each trunk for an offered network load of 18 Erlangs.

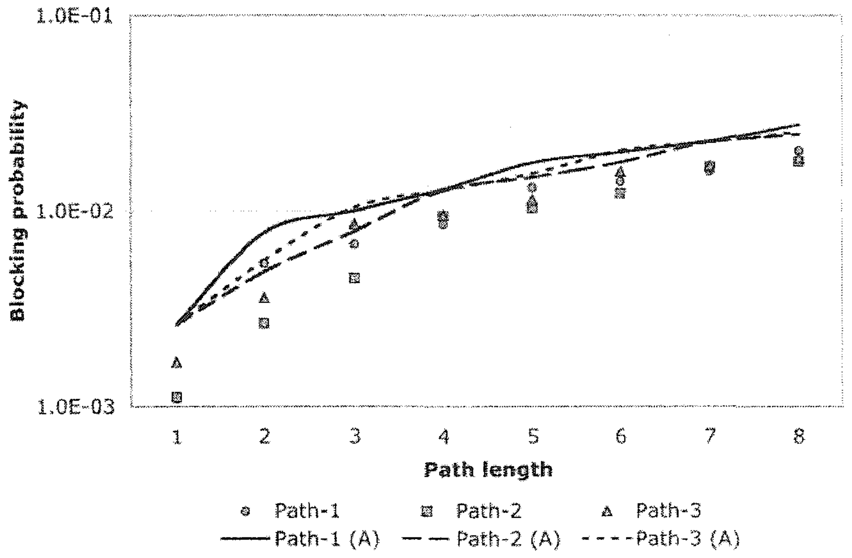


Figure 5.31 Blocking performance of 9-node heterogeneous uni-directional ring network with nodes employing full-permutation switching in each trunk for an offered network load of 21 Erlangs.

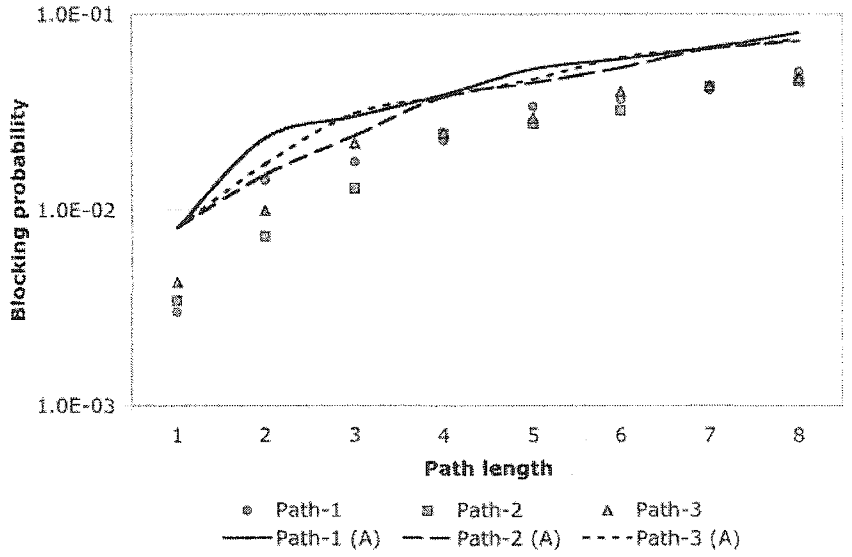


Figure 5.32 Blocking performance of 9-node heterogeneous uni-directional ring network with nodes employing full-permutation switching in each trunk for an offered network load of 24 Erlangs.

2) would experience the least blocking while calls with intermediate node as a wavelength-level grooming node (Path-1) would have the highest blocking among calls that require two-hop connections. Similarly, for three-hop connections, calls with intermediate nodes as FG and TG nodes (Path-2) would experience lowest blocking while calls with TG and WG nodes as intermediate nodes (Path-3) would experience the maximum blocking. Now, note that as more calls requiring connections for two and three hops will be rejected at a wavelength-level grooming node due to the insufficient switching capacity at the immediate neighboring node, calls requiring one-hop connection originating at the WG node would experience lesser blocking performance. It is observed that these performance trends remain the same with the increasing load, thus depending only on where the nodes are positioned. It is to be noted that although the analytical model shows these trends, the difference in blocking performance between calls of different categories are not exactly the same as seen in the simulation results. Hence, minor differences in the blocking probabilities seen through simulations may not be observed through analytical model.

Figures 5.33 through 5.34 show the blocking performance of a 9-node heterogeneous ring network with nodes employing channel-space switching. The configuration of the nodes in the ring is similar to the one considered earlier (shown in Figure 5.27). It is observed that the difference in blocking performance observed through simulation for calls originating in different nodes but having the same path length are as pronounced as observed when full-permutation switching is employed at the nodes. Hence, the analytical model predicts almost the same blocking performance for the paths originating at different nodes but having the same length.

Figures 5.37 through 5.40 show the blocking performance for paths originating at different nodes versus the path length for offered network loads of 72, 78, 84, and 90 Erlangs, respectively, in a 3×3 heterogeneous uni-directional mesh-torus network (as shown in Figure 5.28) with nodes employing full-permutation switching in every trunk. It is observed that the performance trends exhibited by the simulation is also reflected by the analytical prediction, although the difference in blocking performance as predicted by simulation and analysis are different.

Figures 5.41 through 5.44 show the blocking performance for paths originating at different

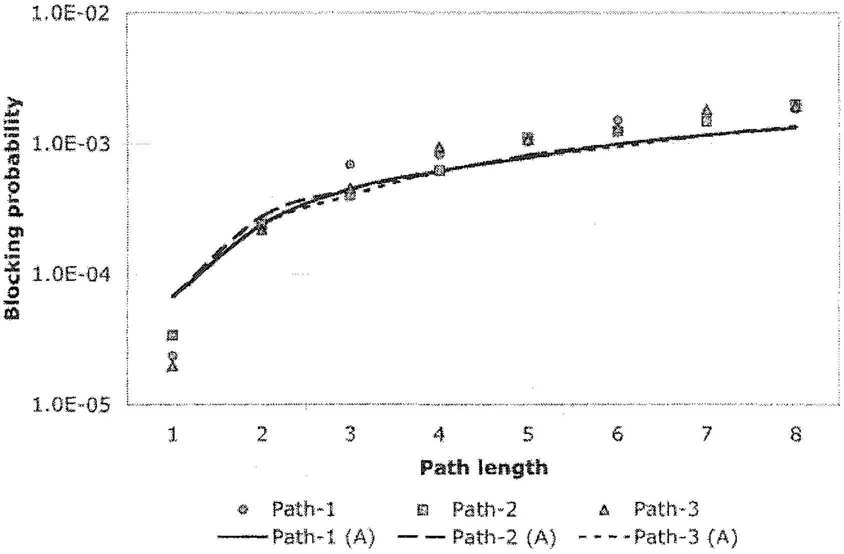


Figure 5.33 Blocking performance of 9-node heterogeneous uni-directional ring network with nodes employing channel-space switching in each trunk for an offered network load of 15 Erlangs.

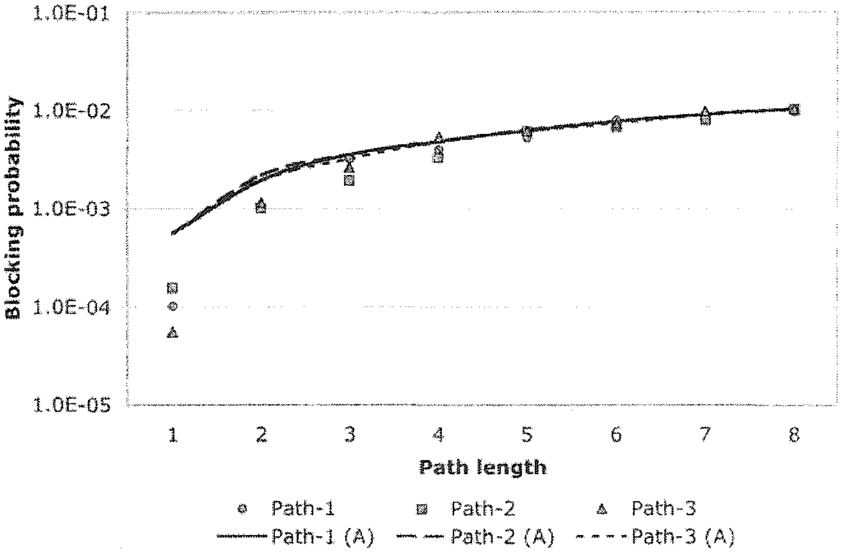


Figure 5.34 Blocking performance of 9-node heterogeneous uni-directional ring network with nodes employing channel-space switching in each trunk for an offered network load of 18 Erlangs.

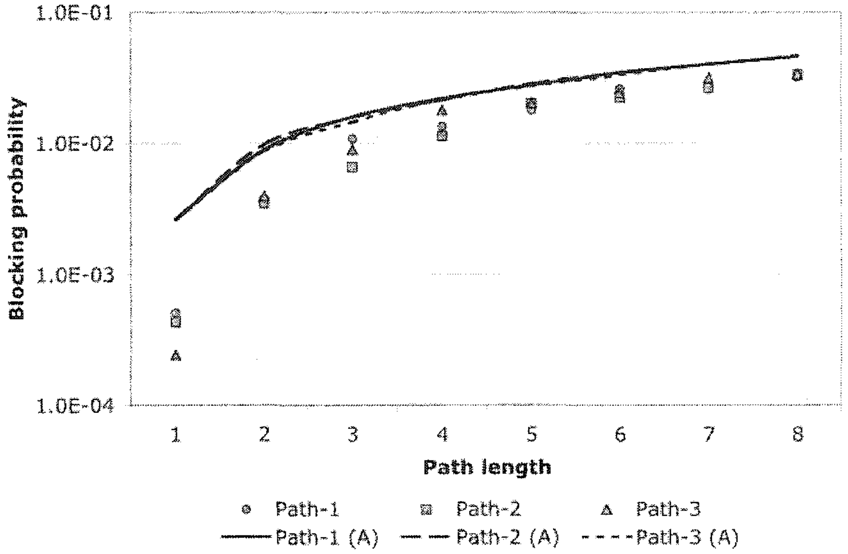


Figure 5.35 Blocking performance of 9-node heterogeneous uni-directional ring network with nodes employing channel-space switching in each trunk for an offered network load of 21 Erlangs.

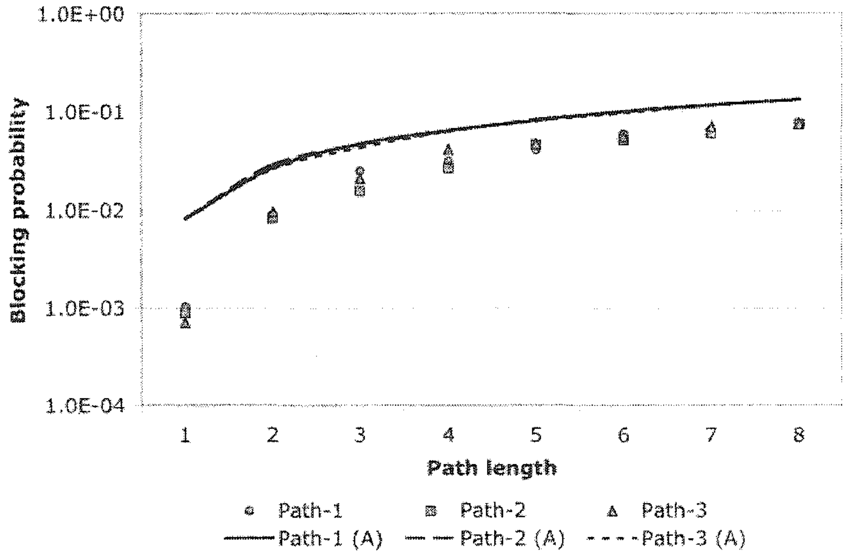


Figure 5.36 Blocking performance of 9-node heterogeneous uni-directional ring network with nodes employing channel-space switching in each trunk for an offered network load of 24 Erlangs.

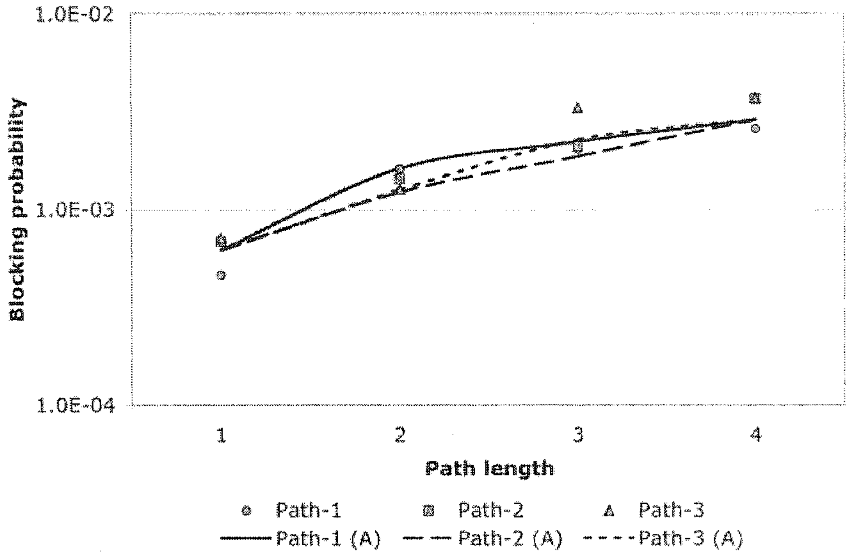


Figure 5.37 Blocking performance of 3×3 heterogeneous uni-directional mesh-torus network with nodes employing full-permutation switching in each trunk for an offered network load of 72 Erlangs.

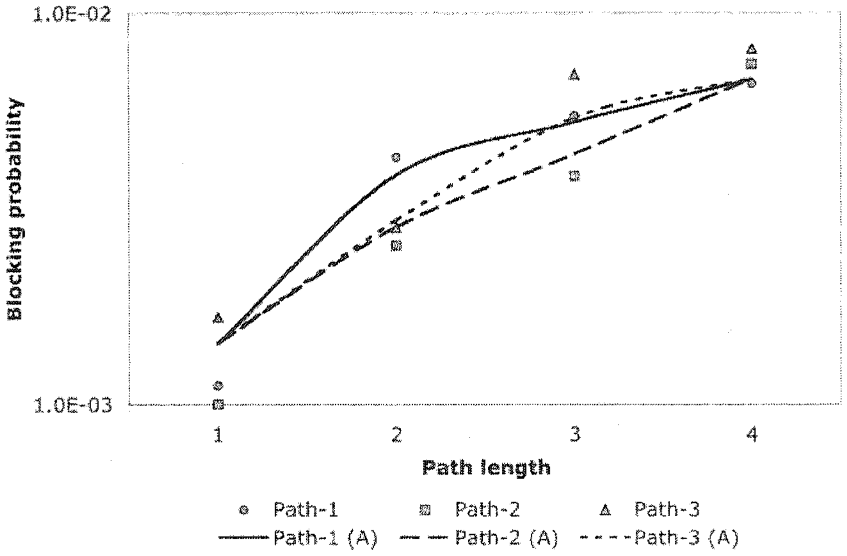


Figure 5.38 Blocking performance of 3×3 heterogeneous uni-directional mesh-torus network with nodes employing full-permutation switching in each trunk for an offered network load of 78 Erlangs.

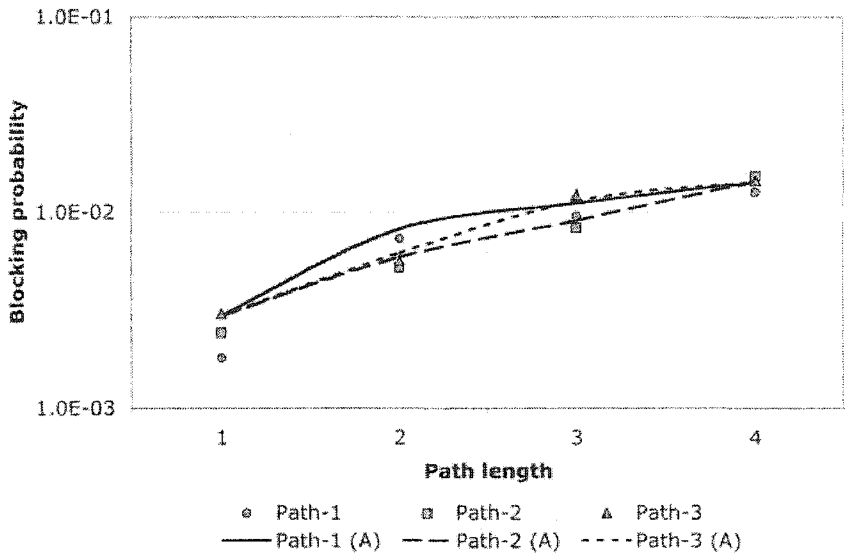


Figure 5.39 Blocking performance of 3×3 heterogeneous uni-directional mesh-torus network with nodes employing full-permutation switching in each trunk for an offered network load of 84 Erlangs.

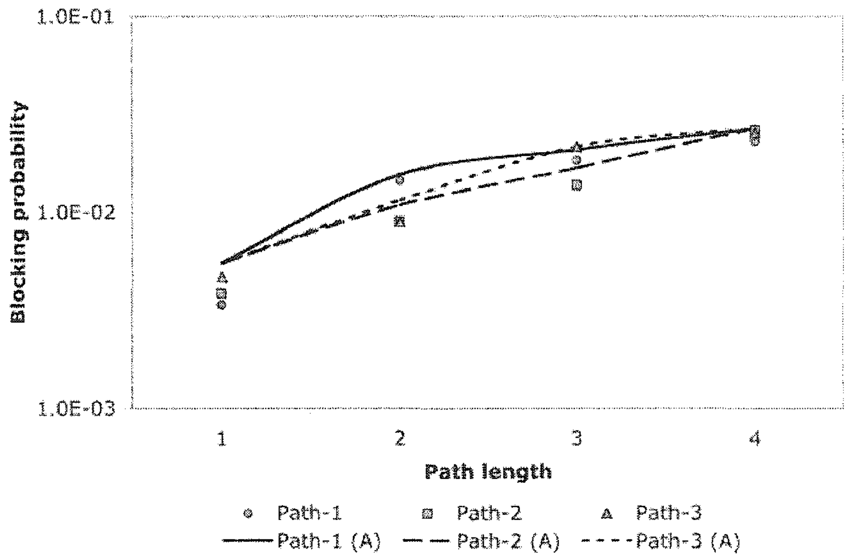


Figure 5.40 Blocking performance of 3×3 heterogeneous uni-directional mesh-torus network with nodes employing full-permutation switching in each trunk for an offered network load of 90 Erlangs.

nodes versus the path length for offered network loads of 72, 78, 84, and 90 Erlangs, respectively, in a 3×3 heterogeneous uni-directional mesh-torus network (as shown in Figure 5.28) with nodes employing channel-space switching in every trunk. It is observed that the difference in blocking probability among calls originating at different nodes but having same path length is not as pronounced as exhibited when full-permutation switching is employed.

Note that the observations made in regard to the difference in blocking probabilities of calls originating at different nodes but having the same path length exhibiting significant differences in the case of full-permutation switching as compared channel-space switching cannot be generalized for any arbitrary arrangement of nodes in the ring.

The important observation to be made in the case of heterogeneous networks is that connections with same hop length could see different blocking probabilities depending on the switching capability of the intermediate nodes. Therefore, it becomes critical to evaluate the importance of a node when the network is upgraded. For example, if only a few wavelength converters are available, then it is critical which nodes in the network are upgraded with this additional flexibility. Determining the criticality of a node would imply evaluating the blocking performance of paths that pass through it under different grooming scenario. The analytical model that is proposed here would allow one to predict the blocking performance of the path with more then one switching architecture. Resource placement algorithms that depend on evaluating path blocking probabilities to identify an optical placement of resources in the network could employ the proposed analytical model. Algorithms for placement of wavelength converters in a wavelength-routed network can be found in [48] and [49]. There has not been significant research in a similar topic in the context of WDM grooming networks.

5.4 Improving accuracy of the analytical model

We demonstrate the use of iterative procedure of the analytical model in order to improve the accuracy of the results by considering an example. We consider a 25-node homogeneous ring network with nodes employing full-permutation switching in every trunk. We assume that all calls

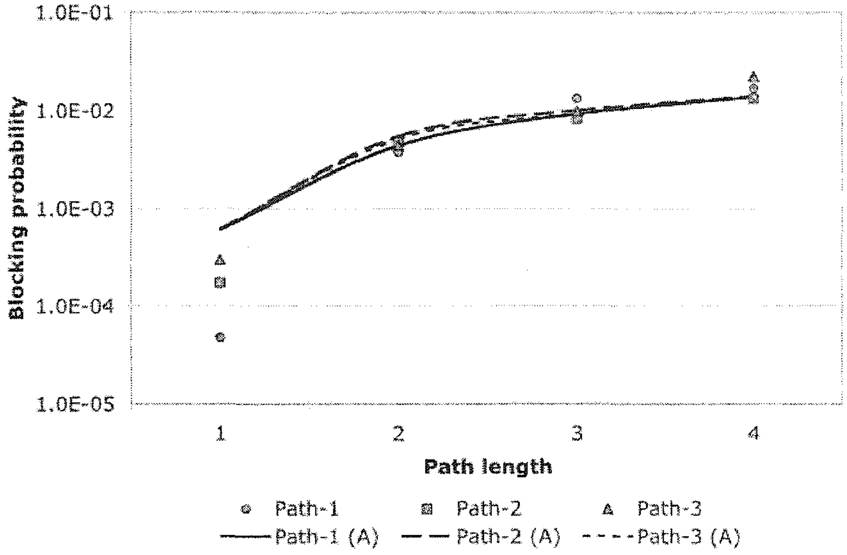


Figure 5.41 Blocking performance of 3×3 heterogeneous uni-directional mesh-torus network with nodes employing channel-space switching in each trunk for an offered network load of 72 Erlangs.

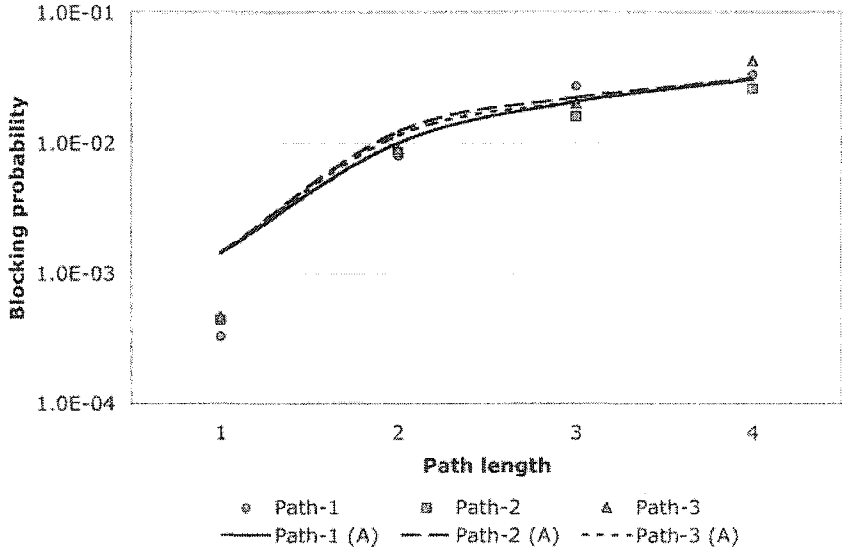


Figure 5.42 Blocking performance of 3×3 heterogeneous uni-directional mesh-torus network with nodes employing channel-space switching in each trunk for an offered network load of 78 Erlangs.

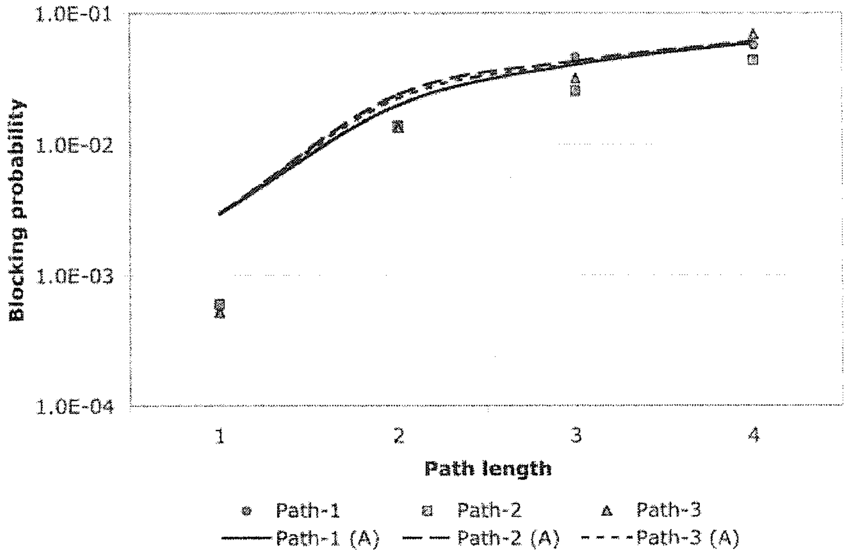


Figure 5.43 Blocking performance of 3×3 heterogeneous uni-directional mesh-torus network with nodes employing channel-space switching in each trunk for an offered network load of 84 Erlangs.

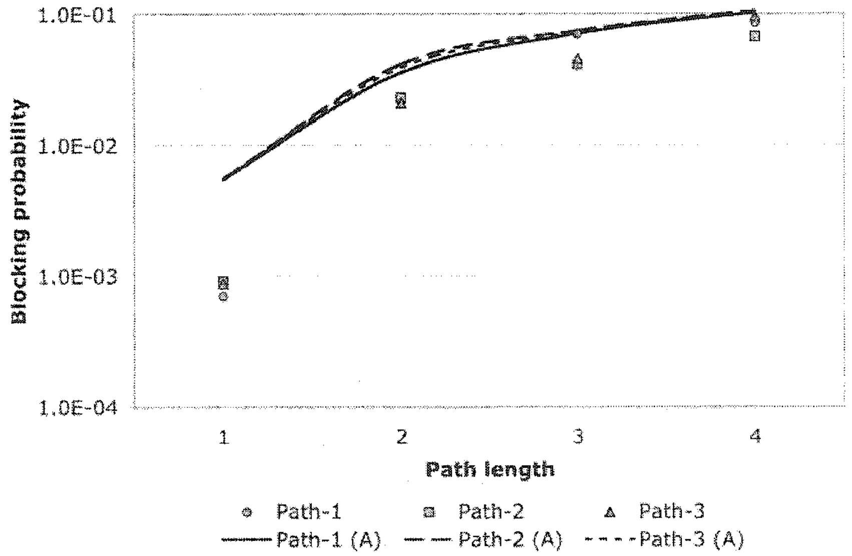


Figure 5.44 Blocking performance of 3×3 heterogeneous uni-directional mesh-torus network with nodes employing channel-space switching in each trunk for an offered network load of 90 Erlangs.

in the network require one time slot capacity.

Figures 5.45 and 5.46 show the blocking probability obtained using simulation and the analytical model. The analytical model is employed twice to get the set of lines marking the lower values. Lines denoted by K×S-FP (A) denote the blockign performance obtained by employing the analytical value once, while those denoted by K×S-FP (2) denote the blocking performance obtained by incorporating the corrected values of link load and correlation values based on the values obtained in the first run of the analysis. It is observed that at low loads (60 Erlangs), blocking experienced by calls requiring twelve hops are under one percent. Hence, the link load and correlation values are not affected significantly to give any significant reduction in employing the analytical model twice. At loads of 100 Erlangs, it is is observed that employing the analytical model for the second time with the corrections to the link load results in obtaining a lower estimate of the blocking performance. Also, note that trend observed in the blocking performance obtained after the second run of the analytical model reflects that observed in the simulation.

Such an iterative procedure can be employed successively by incorporating changes to link load and correlation obtained from the previous run. However, networks are typically designed to have blocking probabilities that are less than 10%. For such scenarios, significant insights into the working of the different switch architectures can be obtained by employing the analytical model twice.

5.5 Summary

In this chapter, we evaluated the blocking performance of WDM grooming networks and compared it against that obtained using simulation. It is observed that the analytical model closely approximates the simulation values. In networks where every request is for one time slot, it is observed that a significant performance gain is obtained with only a channel-space switch as compared to a full-permutation switch. Channel-space switch architecture has fewer number of switching stages as compared to full-permutation switch architecture. It is also observed that the blocking performance of tree establishment can be approximated by blocking performance of a path hav-

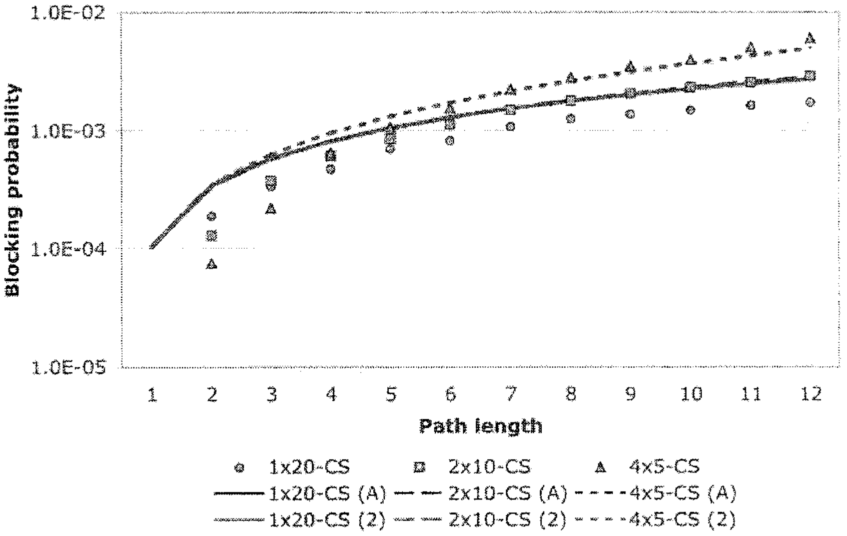


Figure 5.45 Blocking performance for establishing calls requiring one time slot in a 25-node homogeneous ring network with nodes employing full-permutation switching per trunk. The values obtained using the analytical model once are denoted by lines with labels marked (A) while the values obtained using the second run are denoted with (2).

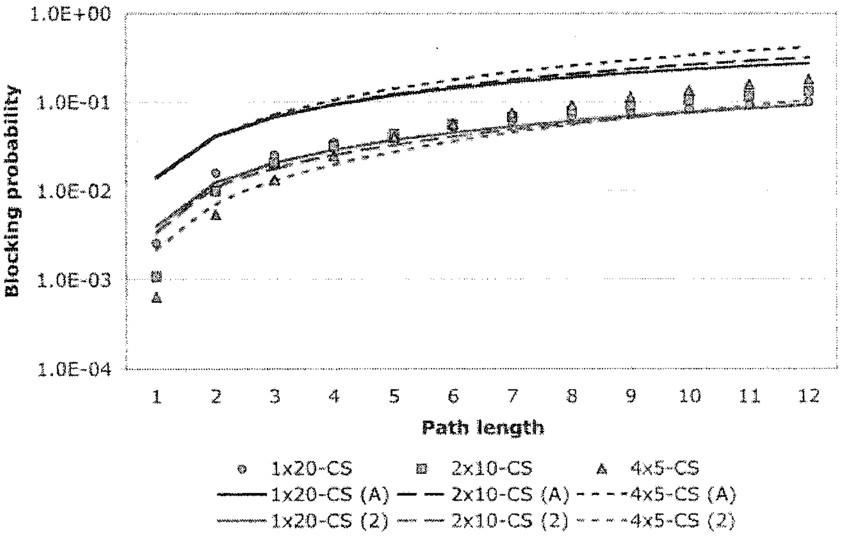


Figure 5.46 Blocking performance for establishing calls requiring one time slot in a 25-node homogeneous ring network with nodes employing full-permutation switching per trunk for an offered network load of 60 Erlangs. The values obtained using the analytical model once are denoted by lines with labels marked (A) while the values obtained using the second run are denoted with (2).

ing the same number of links as that of the tree. Under multi-rate traffic scenario, where every request can have multiple time slot requirement and all the time slots have to be assigned within the same trunk at every node, it is observed that increasing the grooming capability at nodes has significant advantages. Techniques such as dispersity routing, where multiple time slot requests are split into multiple single time slot requests and assigned channels independently on a chosen path, could be employed to improve network performance under restricted grooming capability. In networks with heterogeneous node architectures, it is observed that the blocking probability of requests depends on the grooming capability of intermediate nodes. It is also observed that the blocking performance trends for different paths remain the same for varying network loads. The iterative technique developed as a part of the analytical model has been shown to improve the accuracy of the analytical model. For typical network operating loads where the blocking probability is expected to be less than 0.1, significant insights on the blocking performance can be obtained by employing the analytical model twice (once with the iteration.).

CHAPTER 6 Dynamic Routing in WDM Grooming Networks

We consider dynamic routing schemes for WDM grooming networks in this chapter. Present day networks realize traffic grooming in the electronic domain. Grooming sub-wavelength traffic in the electronic domain has one significant advantage – Wavelength Conversion comes free. But the penalty is paid in terms of knowing the precise data format of the incoming traffic at intermediate nodes. Considering that traffic grooming has gained significance in the recent past, the present-day technology that is employed for traffic grooming is still in its days of infancy. Clearly, as the technology matures and the networks grow, it will be impractical to expect intermediate nodes to understand every protocol that could potentially be used in these networks. Hence the future networks, at least in the core, are expected to employ transparent switching functionalities.

Upgrading the existing networks in the near future is less likely to see a quantum leap in the employed technology. Hence, the first generation of all-optical grooming solution would have limited functionality, such as wavelength-level grooming. Also, it becomes increasingly necessary to quantify the benefits of having sophisticated switching technology in the core networks to justify the cost of deployment. Hence, in this chapter, we focus our research in identifying the benefits of employing different routing schemes and the impact of increasing grooming capability.

We assume a wavelength-level grooming network to study the performance of different routing algorithms. Wavelength-level grooming networks fall into the class of homogeneous TSNs, where every node views a wavelength as a trunk. Hence, in a network with W wavelengths per link, the link information matrix for a link would be a $W \times W$ diagonal matrix. For easy understanding, we simplify the notation from a matrix representation to a vector representation.

Consider the example network shown in Figure 6.1. Assume that every link carries 2 wavelengths with 4 timeslots per wavelength. The figure shows the available wavelength capacity (in timeslots) on the two wavelengths on each of the links at some point of time during the network operation. We illustrate the different routing algorithms discussed in this chapter using this example network.

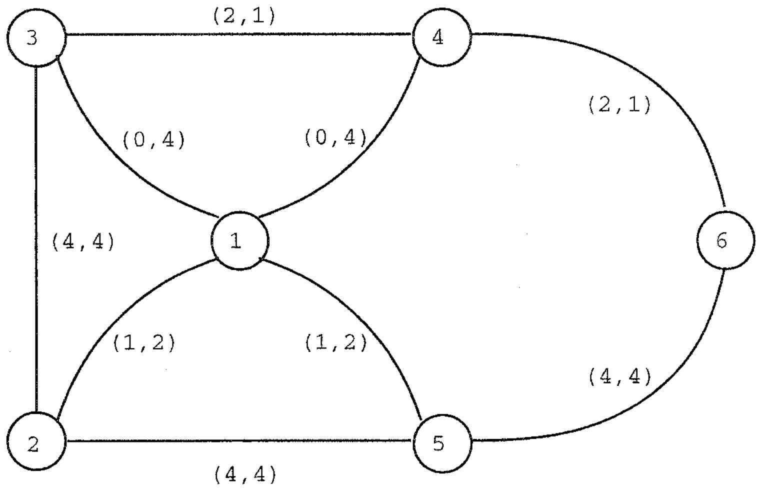


Figure 6.1 Example network employing two wavelengths per fiber. The tuples denote the available capacity on the two wavelengths.

6.1 Information collection

Every node in the network is assumed to maintain the global state information through a link-state protocol. The information collection and path selection are based on two metrics: available wavelength capacity and hop length. A path with W wavelengths with C channels per wavelength is denoted by a vector $\{(A_1, A_2, ..., A_W); H\}$, where each A_w $(1 \leq w \leq W)$ denotes the number of available channels on a wavelength and H denotes the hop-count. For a link vector, the value of H is 1. The available capacity on a wavelength w and hop length of a path p is denoted by A_p^w and H_p respectively. It is to be noted that the link information here is represented in a vector form for simplicity.

Traditional Dijkstra's shortest path algorithm is extended to the above link-state vector, referred to as extended Dijkstra's shortest path (EDSP) algorithm, and is employed at every node in the network. The EDSP algorithm uses the link-state vector as defined above instead of a single metric that is traditionally used. The EDSP algorithm has two important operations: (1) combining two path vectors and (2) selecting the best path vector. Let ψ_{ik} and ψ_{kj} denote the path vectors from node i to k and from node k to j, respectively. The path vector from node i to j through k is obtained by combining the path vectors ψ_{ik} and ψ_{kj}, denoted by $\psi_{ij} = \psi_{ik} \oplus \psi_{kj}$. The vectors are combined in different ways depending on the grooming capability of the node k.

The second operation of selecting the best path vector from a given set of path vectors is defined by a specific path selection policy. For example, the traditional shortest path algorithm selects a path with minimum hop length.

6.1.1 Wavelength-level grooming networks

In wavelength-level grooming networks, connections cannot be switched from one wavelength to another. Hence, wavelength continuity constraint is obeyed. Two paths vectors ψ_{ik} and ψ_{kj} are combined at a WG node to obtain ψ_{ij} where $A_{ij}^{w} = min(A_{ik}^{w}, A_{kj}^{w})$ and $H_{ij} = H_{ik} + H_{kj}$.

Consider the example network in Figure 6.1. Assume that Node 4 can perform wavelength-level grooming. The path from Node 1 to 6 through 4 is described by the vector $\psi_{16} = \{(0, 1); 2\}$.

6.1.2 Sparse full-grooming networks

In sparse full-grooming networks, a few nodes in the network have full-grooming capability. Low-rate traffic streams can be switched across wavelengths at these nodes. Hence, the maximum capacity of a connection that can be switched by an FG node corresponds to the maximum available capacity across different wavelengths on an output link. In such a scenario, the available capacity on a path P_{ij} on a wavelength w is obtained by combining two path metrics P_{ik} and P_{kj} as $A_{ij}^{w} = min(A_{ik}^{w}, A_{kj}^{max})$ where A_{kj}^{max} denotes the maximum available capacity across different wavelengths on the path from k to j. The hop length is computed as $H_{ij} = H_{ik} + H_{kj}$.

Consider the example network in Figure 6.1. Assume that full-grooming is available at Node 4. The vector for the path from Node 1 to 6 through 4 is obtained as $\psi_{16} = \{(0,2);2\}$.

6.1.3 Constrained grooming networks

In constrained grooming networks, grooming is accomplished only on the dropped wavelengths. Consider the example in Figure 6.1. Let two connections exist between nodes 3 and 6 through 4. Assume that the first connection occupies two channels on the first wavelength while the second occupies three on the second wavelength. Although both the wavelengths have free channels, they cannot be used to reach Node 4 as the wavelengths are not dropped at Node 3. Hence, when a lightpath is setup between a source and destination, they can be treated as logical neighbors. The established lightpaths can then be used to route further connections by updating the link-state information.

If the nodes in the network shown in Figure 6.1 perform constrained grooming, then the network is viewed as shown in Figure 6.2. The lightpaths that are established between nodes that are not physical neighbors are shown in dotted lines. Two path vectors in such networks are combined in a manner similar to that of wavelength-level grooming networks.

6.2 Path Selection Algorithms

The path selection algorithms considered in this dissertation are restricted to destination-specific approaches. The different path selection algorithms specifies the rule for selecting the best path vector in the EDSP algorithm. Four examples of path selection algorithms are listed below:

Widest-Shortest Path Routing (WSPR): In this approach, the available wavelength capacity vector on a path is ordered in descending values of the individual wavelength capacities. Thus an available wavelength capacity vector $A'_p = (A'_1, A'_2, ..., A'_W)$ is said to be ordered descending if $A'_i \geq A'_j$ for $i < j$ and $1 \leq i, j \leq W$.

An ordered vector $A' = (A'_1, A'_2, ..., A'_W)$ is said to be smaller than another ordered vector $B' = (B'_1, B'_2, ..., B'_W)$ if for some i $(1 \leq i \leq W)$, $A'_i < B'_i$ and for all $j < i$ $A'_j = B'_j$. The vectors

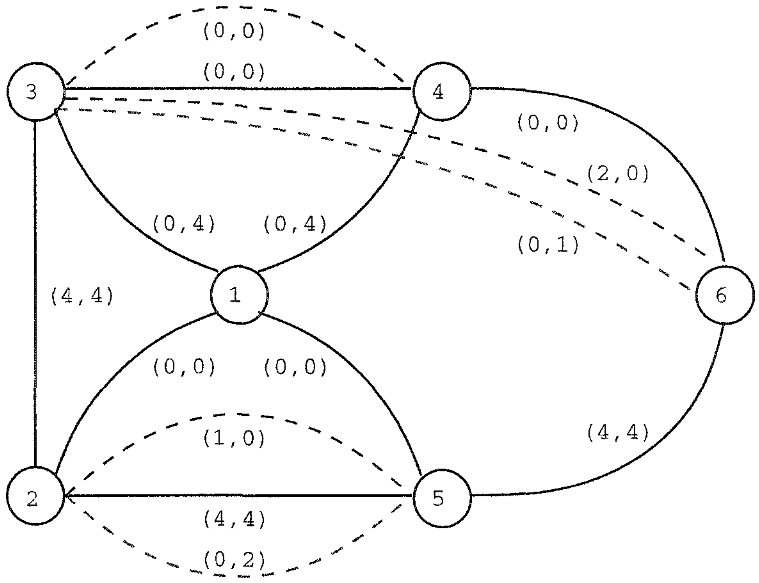


Figure 6.2 Visualizing a constrained grooming network.

are said to be equal if $A'_i = B'_i$, for all i, where $1 \leq i \leq W$. Otherwise, A is said to be larger than B. A path with the largest path-vector is said to be the *widest* path and is chosen for establishing a connection. In case of a tie, the path with the minimum hop length is chosen.

Shortest-Widest Path Routing (SWPR): This is the conventional shortest-path routing based on the hop-length. If more than one such path is available, the widest among them is chosen.

Available Shortest Path Routing (ASPR): In this approach, the shortest path among those that can accommodate the request is chosen. The paths that can accommodate the request are those that have at least one wavelength that can accommodate the request. If two paths that can accommodate the request have same hop length, then one of them is chosen at random.

If two path vectors are equal by any of the above algorithms, one of the path vectors is chosen at random. Note that only the selection of the path vector is based on the ordered available capacity vector. WSPR and SWPR are examples of destination-specific routing schemes while ASPR is an example of request-specific routing. In ASPR, the set of feasible paths is chosen based on the capacity requirement of the request.

6.2.1 An example

Consider the example network shown in Figure 3.1. Assume that every link carries two wavelengths and each wavelength is divided into 4 time slots. The tuples shown in the figure correspond to the available wavelength capacity on each wavelength.

Consider a request that originates from Node 2 destined to Node 6. The SWPR algorithm selects the path $2 \rightarrow 5 \rightarrow 6$ with path vector {(0,0);2}. The request cannot be accommodated due to lack of capacity on link $5 \rightarrow 6$. The WSPR algorithm selects the path $2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 6$ with path vector {(0,4);4}. These paths are chosen irrespective of the request requirements.

The ASPR algorithm selects the path based on the request. If the request is for one time slot, then the path $2 \rightarrow 1 \rightarrow 4 \rightarrow 6$ with a path vector {(0,1);3} or $2 \rightarrow 3 \rightarrow 4 \rightarrow 6$ with path vector {(2,2);3} is chosen. If the request is for 2 time slots, the path $2 \rightarrow 3 \rightarrow 4 \rightarrow 6$ with path vector {(2,2);3} is chosen. If the request is for 3 or 4 time slots, then the path $2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 6$ with path vector {(0,4);4} is chosen.

6.2.2 Dispersity Routing

If the connection for a request of capacity b has to be established only on one wavelength, then SWP and WSP algorithms are used. If multiple wavelengths can be used to meet the capacity requirement, the request is split into b requests of unit capacity each. If the path from the source to the destination can accommodate the set of b requests, then the request is said to be accepted. Otherwise, it is blocked. Such an approach to routing a larger capacity requests by splitting into smaller capacity requests is called *dispersity routing*. In this chapter, it is assumed that a request can be assigned channels that are dispersed over wavelengths of the same path, referred to as *wavelength-level dispersity routing*. When dispersity routing is employed, a path is said to be wider if the total available capacity on all the wavelengths in the path is higher.

6.3 Performance Evaluation

The performance of four path selection algorithms described in the previous section are evaluated on the NSFnet network. The 14-node 22-link NSFnet network is shown in Figure 6.3. The performance results reported in this section are restricted to wavelength-level grooming employed at all nodes in the network.

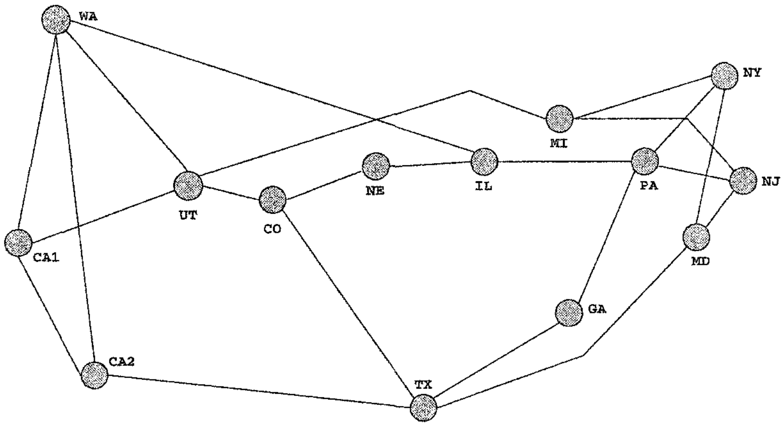


Figure 6.3 The NSFNET network.

When a request arrives at a node, the path to the destination is chosen using one of the above mentioned path selection schemes. The wavelength allocated to establish the connection is the one that can just accommodate the capacity of the request (best-fit wavelength assignment). SWPR and WSPR algorithms are used in networks that do not allow dispersity routing while SMSPR and MSSPR are used by networks that employ dispersity routing.

6.3.1 Experimental setup

The experimental setup for the simulation is based on the following assumptions.

- The arrival of requests at a node follow a Poisson process with rate λ and are equally likely to be destined to any other node.
- The holding time of the requests follow an exponential distribution with unit mean.

- The capacity requirement of a request is equally likely to take integer values from 1 to 8.
- Every link has 128 channel capacity divided over W wavelengths.

A network with links having 16 wavelengths and 8 channels per wavelength are referred to as 16×8 network. Four different wavelength-channel combinations are considered: (1) $16{\times}8$ (2) $8{\times}16$ (3) $4{\times}32$ and (4) $1{\times}128$. The requests are generated independently at a rate of $N\lambda$, where N denotes the number of nodes in the network. The requests are equally likely to have any of the N nodes as its source. The generated requests are fed to the different networks running in parallel and their performances are measured. A total of 6×10^5 requests were generated with performance metrics being measured in batches of 10^5 requests. The average of the performance metrics over observed six set of values are reported in the results.

6.3.2 Performance metrics

The performance metrics that are measured are the request blocking probability, average path length of an accepted connection (Z), average shortest-path length of an accepted request (Z_m), and network utilization (η).

The blocking probability is computed as the ratio of the number of blocked requests to the number of total requests generated. Z is computed as the average of the length of the paths assigned to the accepted requests by a specific routing algorithm. Z_m is computed as the average of the shortest-path length of the requests accepted by the routing algorithm. It can be observed that SWPR would have $Z = Z_m$ while other routing schemes would have $Z \geq Z_m$.

The *network utilization* is computed by assigning an effective network capacity requirement for a request. A request r for capacity b from source s to destination d has an effective capacity requirement of $b \times h_s$, where h_s is the shortest path length from the source to the destination. This effective capacity requirement of a request is the minimum capacity that is required in the network to support the request, irrespective of the routing algorithm. If a routing algorithm selects a path of length h for the connection, $b(h - h_s)$ denotes the additional capacity used by the network to support the connection.

The effective network capacity utilized at an instant of time, denoted by U, is defined as the sum of the effective network capacity requirement of all the connections that are active at that instant. The value of U at any instant of time is bounded by $L \times C$, where L is the total number of links in the network and C is the capacity on each link. The network utilization is then computed as the ratio of the effective used capacity to the maximum capacity of the network as $\eta = \frac{U}{L \times C}$.

6.3.3 Effect of routing algorithms

Figure 6.4 shows the blocking performance of different routing algorithms on 16×8 and 1×128 NSFnet networks. It is observed that ASPR performs better than SWPR and WSPR. It is also observed that as the network load is increased, the blocking performance of WSPR worsens as it routes connection over wider but longer paths resulting in wastage of bandwidth.

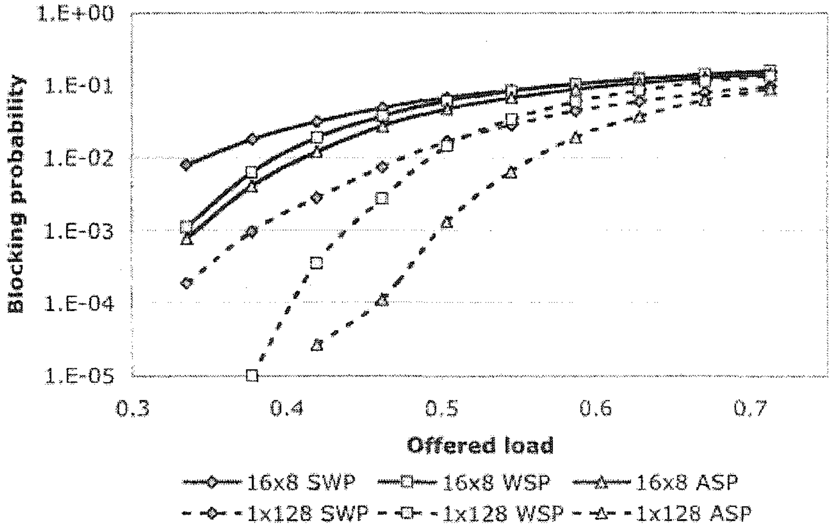


Figure 6.4 Blocking performance of different routing algorithms on a 16×8 and 1×128 NSFnet networks.

Figure 6.5 shows the network utilization under different routing algorithms on 16×8 and 1×128 NSFnet networks. It is observed that ASPR achieves the the maximum utilization compared to WSPR and SWPR.

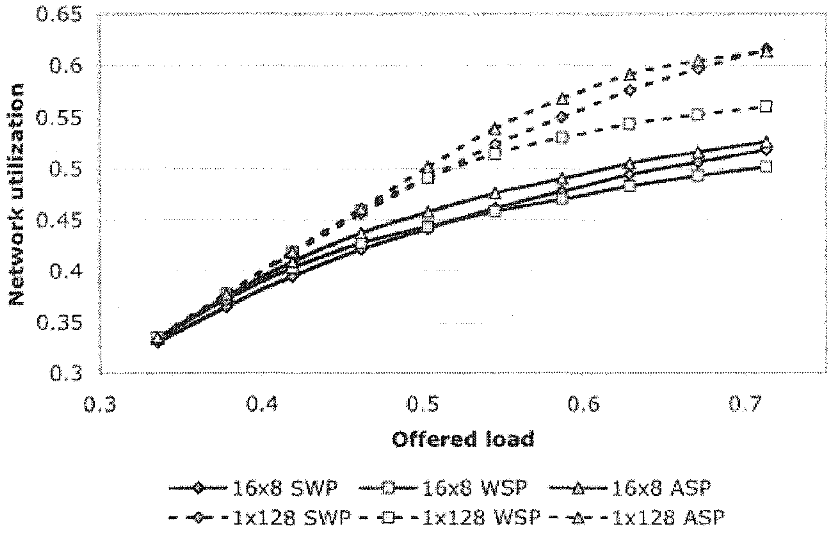


Figure 6.5 Network utilization of different routing algorithms on a 16×8 and 1×128 NSFnet networks.

More insights into the working of the algorithms are obtained by observing the average path length of the connections established. Figure 6.6 shows the average path length of connections established in the networks by different routing algorithms. It is observed that WSPR selects longer paths for establishing connections as compared to ASPR and SWPR. This difference is significant when the grooming capability in the network is increased. This indicates that increasing the grooming capability helps dynamic routing algorithms in finding more paths but at the expense of longer path lengths. SWPR has the least value for this metric as it selects only shortest paths.

The average path length for a connection established under WSPR remains almost a constant with load as the preference is given to distributing the load in the entire network. On the other hand, SWPR attempts only on the shortest path. As network load increases, more longer path requests are blocked, hence results in a decrease in the average path length. ASPR behaves similar to SWPR under low loads. However, as the offered load to the network is increased, ASPR attempts to route connections on the longer paths, hence the trend of increasing average path length with increasing offered network load.

The average path length all the routing algorithms for 1 × 128 network is higher than that of

16 × 8 network because more requests are accepted, but along longer paths in the former network due to the increased grooming capability. Increasing the grooming capability improves the chances of finding a path between nodes, though it would result in wastage of network resources.

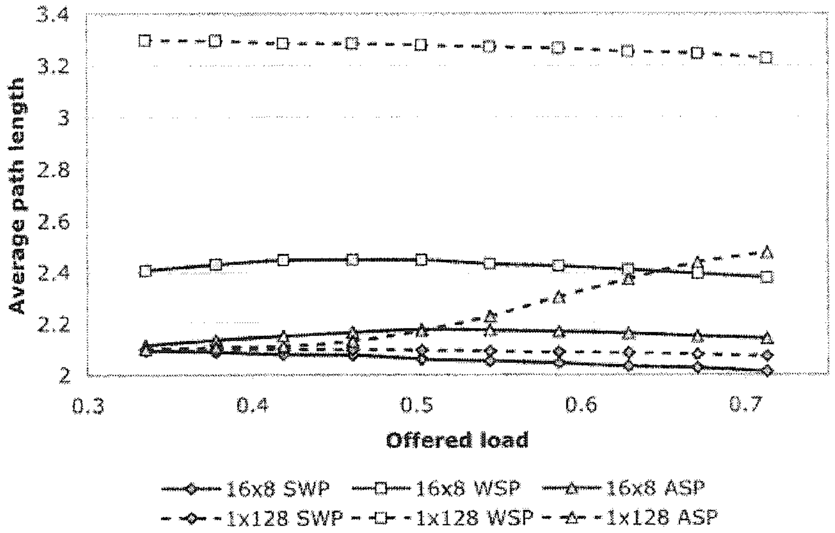


Figure 6.6 Average length of connections established by different routing algorithms on a 16×8 and 1×128 NSFnet networks.

Figure 6.7 shows the average shortest path length of accepted requests for different routing schemes. At low loads, very few requests are rejected. Hence, the average shortest path length of accepted requests is the same for different routing schemes. When the offered load to the network is increased, requests with longer shortest path length experience more blocking resulting in a bias in favor of requests with smaller shortest path length. The lower the value of this metric for a routing algorithm, the stronger is the bias in favor of requests with smaller path length. ASPR performs the best with respect to this fairness metric. It is observed that increasing the grooming capability enhances the performance of the routing schemes with respect to this metric.

The routing schemes also exhibit a bias in favor of smaller capacity connections when the offered load to the network is increased. Requests for larger capacity experience more blocking than the ones for smaller capacity. Such a behavior is pronounced in networks that have lesser grooming capability. Figure 6.8 shows the average capacity of accepted requests for different

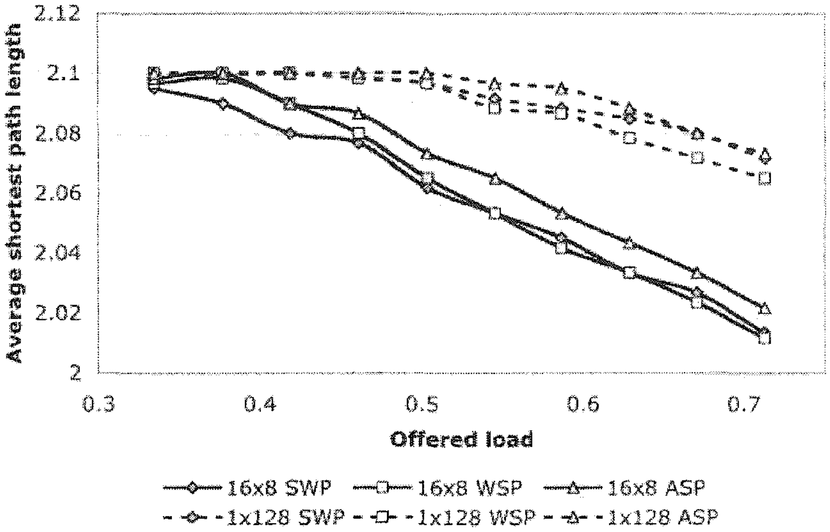


Figure 6.7 Average shortest path length of connections established by different routing algorithms on a 16×8 and 1×128 NSFnet networks.

routing schemes. It is observed that increasing the grooming capability enhances the fairness of the routing algorithms with respect to requests of different capacity requirement.

The average shortest path length and average capacity of accepted requests quantify the fairness property of the routing algorithms. An ideal routing algorithm would have a constant value for these metrics at all network loads.

It is observed that ASPR offer better performance over SWPR and WSPR algorithms with respect to various performance metrics. Similar performance results were obtained for 8×16 and 4×32 NSFnet networks.

6.3.4 Effect of dispersity routing

In order to improve the network performance under different routing algorithms, dispersity routing is also employed. Dispersity routing removes the constraint of routing a connection entirely on a wavelength, hence provides greater flexibility in assigning connections. Figures 6.9 and 6.10 show the performance of different routing algorithms on a 16×8 NSFnet network. It is observed that ASPR performs the best when dispersity routing is employed.

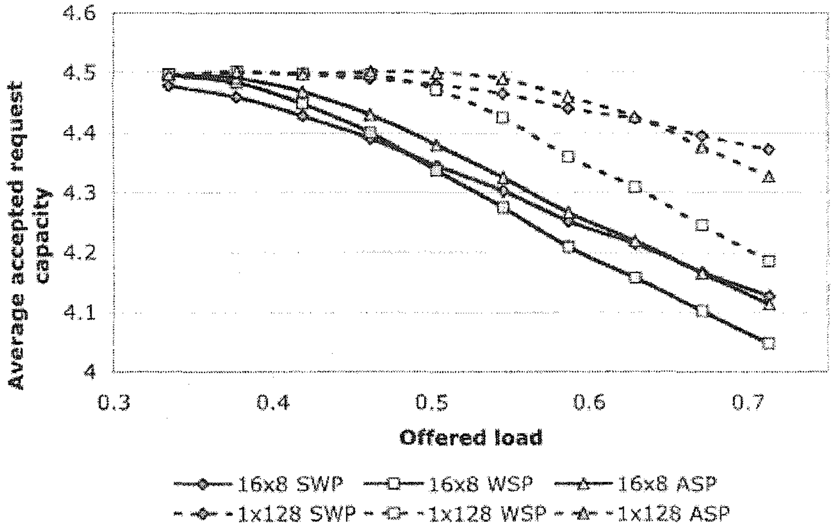
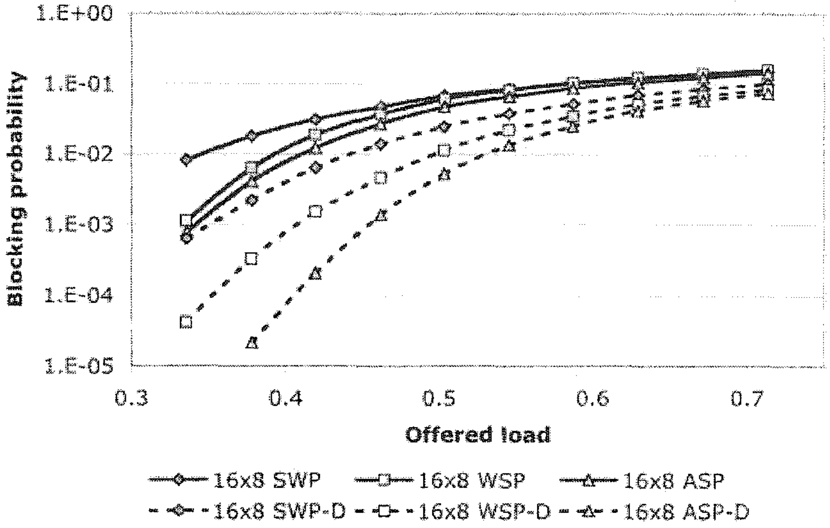


Figure 6.8 Average capacity of accepted requests for different routing algorithms on a 16×8 and 1×128 NSFnet networks.

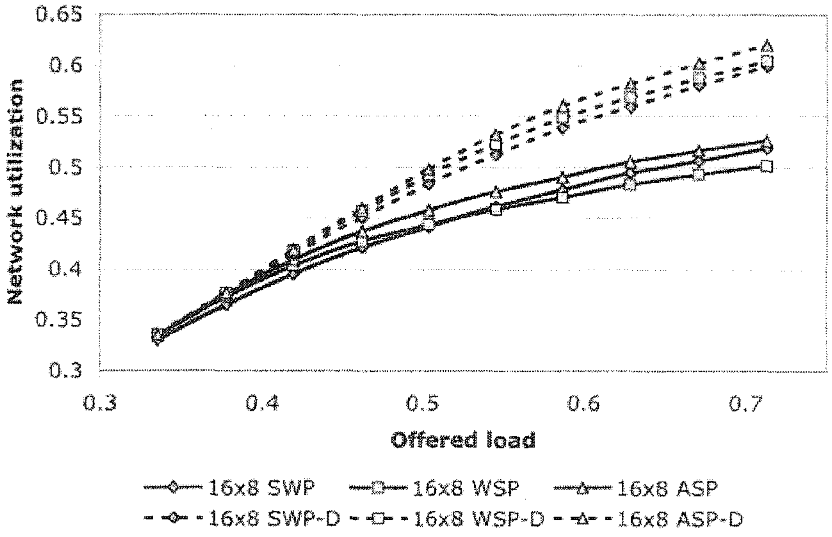
From the results it is clear that employing dispersity routing significantly improves network performance. At offered load of 0.7, a network employing ASPR with dispersity routing achieves an utilization of 0.62 while the utilization achieved by emplying ASPR without dispersity routing is 0.526, a 17.9% improvement. Under dispersity routing, only the end nodes need to maintain the information regarding how the connection is split, while the intermediate nodes would route the connection as if they were unit capacity connections.

6.3.5 Effect of dispersity Vs. grooming capability

While wavelength-level dispersity routing is one mechanism to achieve increased network performance, alternatives to improve grooming capability can also be considered as a solution. For example, instead of employing 16 wavelengths with 8 time slots, one could employ 8 wavelengths and 16 time slots. As call requirements still vary only between 1 and 8 time slots, the latter wavelength and time slot combination would result in reduced blocking. However, such a change increases the transmission speed on a wavelength, requiring faster switches at the nodes. On the extreme, one could consider one wavelength with 128 time slots. In this case, the switching speed

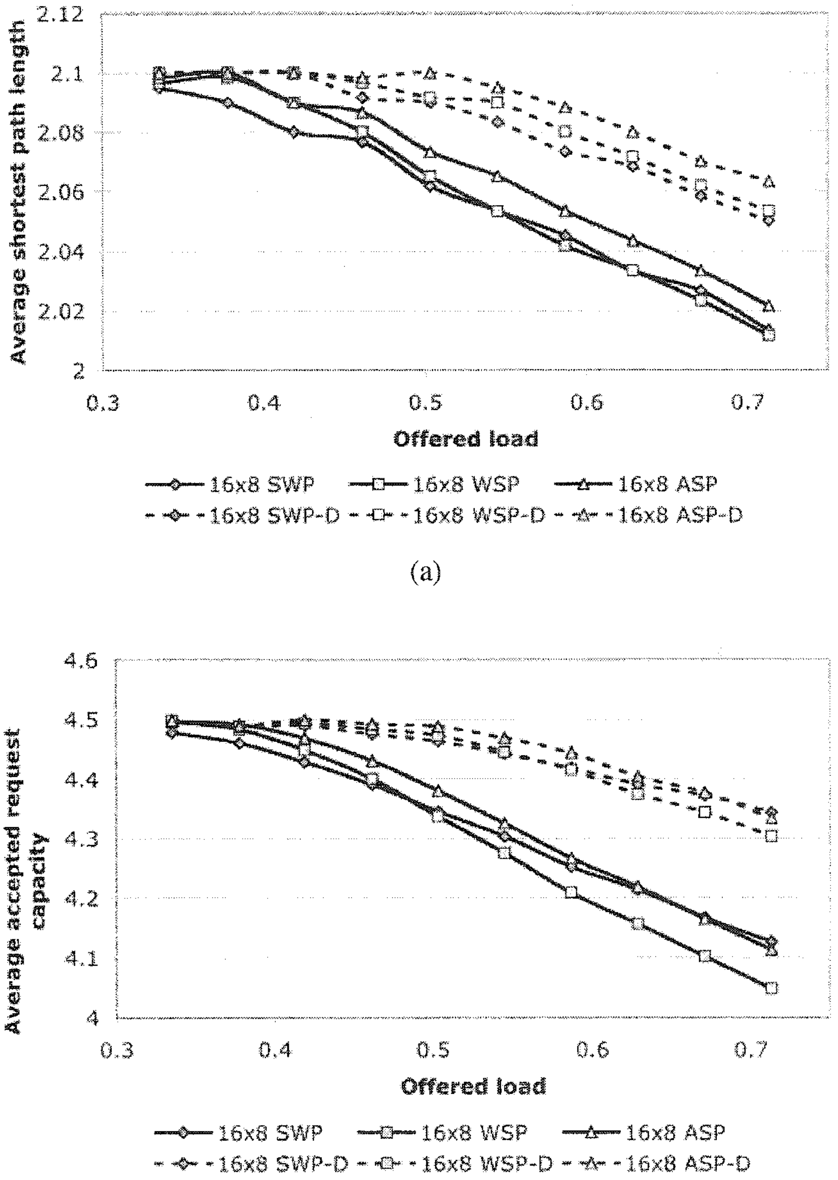


(a)



(b)

Figure 6.9 Performance of different routing algorithms on 16×8 NSFnet network with and without dispersity routing. (a) Blocking probability; (b) Network utilization.



(a)

(b)

Figure 6.10 Performance of different routing algorithms on 16×8 NSFnet network with and without dispersity routing. (a) Average shortest path length; (b) Average accepted request capacity.

has to be 16 times faster as compared to that in a 16-wavelength 8-time slot network.

Another approach to achieve improved performance is to employ multiple wavelengths, but include wavelength conversion capability. This would eliminate the need for faster switches. For example, consider a network with 8 wavelengths and 16 time slots in each wavelength. If limited wavelength conversion capability is provided at a node where wavelength W_1 and W_2, W_3 and W_4, W_5 and W_6, and W_7 and W_8 can be interchanged, then this is similar to a network employing 4 wavelengths and 32 time slots in each wavelength. In such an architecture, the network cost is higher due to the wavelength conversion capability at every node in the network.

We study the performance of dispersity routing and varying grooming capability to evalute the trade-off. We evalute four different wavelength and time slot combinations: 16×8, 8×16, 4×32, and 1×128. In the case of 1×128 there is no distinction between routing a connection with or without dispersity.

Figure 6.11 shows the blocking performance of ASPR with dispersity routing for the four different wavelength and time slot combinations.

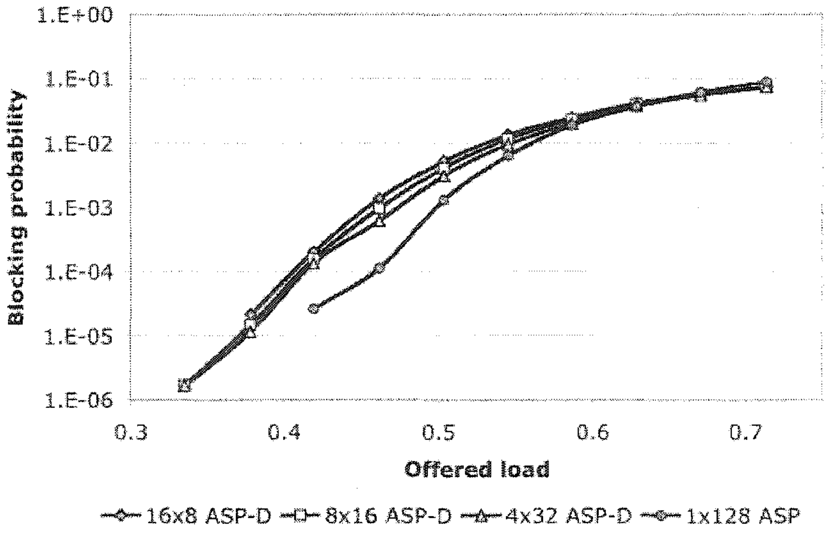


Figure 6.11 Blocking performance of different routing algorithms on NSFnet network with different levels of grooming capability and dispersity routing.

It is observed that the blocking performance is reduced with increasing grooming capability. This performance improvement is observed to be a gradual. It is observed that at offered loads of 0.5 and above, the blocking performance under various grooming capabilities within the same order of magintude. The performance with respect to other metrics such as network utilization, average shortest path length, and average accepted call capacity were found to be very close to distinguish when plotted in graphs, hence are not shown.

It can be concluded from the extensive simulation results shown in this dissertation that a significant performance improvement can be achieved with dispersity routing while having lesser grooming capability.

6.4 Summary

In this chapter, we studied dynamic routing in wavelength-level grooming networks. Three routing algorithms, Shortest Widest Path, Widest Shortest Path, and Available Shortest Path, have been studied using extensive simulations for their performance. A new metric to evaluate routing algorithms that reflects the network utilization is also proposed. We showed that routing connections based on the shortest available path is a preferred approach. We quantified the benefit of routing larger capacity connections by breaking them into multiple smaller capacity streams, referred to as dispersity routing. It is observed that up to 20% increase in network utilization is observed with dispersity routing, similar to that obtained with increasing the grooming capability to the maximum at every node in the network. This result reveals that employing dispersity routing is an attractive alternative compared to increasing the grooming capability at nodes in the network in order to reduce the network cost.

CHAPTER 7 Conclusions

Optical communications employing wavelength division multiplexing (WDM) has emerged as the prominent technology for future high-speed backbone networks. While the transmission speed on a wavelength has increased with advances in technology, bandwidth requirement of users do not increase at the same rate. Hence, the mismatch in the requirement has resulted in various wavelength sharing techniques. One such wavelength sharing technique is time division multiplexing (TDM) over WDM networks. Such networks are referred to as WDM/TDM or WDM grooming networks.

WDM grooming networks have slowly evolved with time from their predecessor wavelength-routed networks. These networks are expected to evolve into an all-optical network. Upgrading existing network to all-optical technology involves both time and money. Financial constraints typically allow only a few nodes in the network to be upgraded. Hence, the nodes in such networks are expected to employ heterogeneous switching architectures. Earlier work in wavelength-routed networks and WDM grooming networks have assumed all the nodes in the networks to employ same switching architectures and cannot be applied or modified easily to model heterogeneous networks.

In this dissertation, we propose a new network model called *Trunk Switched Network* (TSN). The links in the network are viewed as channels. The channels are combined to form groups called *trunks*. With this concept, we develop a node architecture that has the capability of switching channels within a trunk. We demonstrate WDM grooming networks employing different grooming architectures can be easily modeled using the proposed network model. We develop a framework for connection establishment in such networks to establish a circuit of a specific bandwidth from a source to destination. We employ a novel matrix-based approach to store information about a link and a generalized matrix multiplication procedure to combine the link information to obtain information on a path. Various path selection algorithms are then developed based on the information collected on the path. We complete the framework by developing a channel assignment methodology from which different channel assignment strategies can be derived. We also demonstrate the usefulness of the network model and connection establishment framework to model specific switching architectures at a node.

We analyze a TSN for its blocking performance. The analytical model is based on the computation of path blocking performance and takes into account the link load correlation in the network. We analyze the blocking performance for establishing a multicast tree by splitting a tree into multiple paths and combining the path blocking probabilities. The analytical model also incorporates requests with multiple channel capacity. We validate the accuracy of the analytical model through extensive simulations. The analytical model assumes a fixed-path routing strategy for reasons of tractability.

We evaluate the performance of dynamic routing algorithms in WDM grooming networks that employs path selection based on the up-to-date network information. We propose a new request-specific routing algorithm called *Available Shortest Path* algorithm that aims at minimizing the network resources used by a connection. The path is selected based on the characteristics of the request as opposed to destination-specific mechanisms where a path is chosen before a request arrival. We demonstrate that employing request-specific routing strategies can significantly improve network performance. We consider dispersity routing approach where a request with large capacity requirement is broken down into multiple requests smaller capacity requirements. We show that dispersity routing could be employed as an alternative to the high-cost solution of increasing the grooming capability at the nodes to improve network performance.

The network model proposed in this paper allows for modeling heterogeneous switching architectures in the network. The model could be employed in the design of optical networks that assume static traffic demands. Such design problems are modeled as optimization problems. The current research in the area of design of wavelength-routed optical networks assume the same switching architecture at all nodes, refer to [50] and references there in. All the nodes in the network are either assumed to employ full-wavelength conversion or no wavelength conversion. Such restrictions can be removed by employing the network model proposed in this dissertation.

The major limitation of the proposed network model is that it restricts the switching of channel only among the same trunk. However, some switching architectures cannot be effectively modeled in the proposed framework, networks with limited range wavelength conversion for example. The proposed network model is a conceptual view of a network and does not depend on technological aspects. Hence, the applicability of the model reaches beyond optical networks as well. The model is applicable in general to networks that employ multiplexing in more than one domain, like combinations of time, frequency/wavelength, code, phase, etc. and employ switching within a subset of channels.

BIBLIOGRAPHY

[1] W. Goralski, *SONET*, McGraw-Hill, second edition, 2000.

[2] R. Ramaswami and K. Sivarajan, *Optical Networks: A Practical Perspective*, Morgan Kaufmann Publishers, 1998.

[3] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WANs," *IEEE Transactions on Communications*, vol. 40, pp. 1171–1182, July 1992.

[4] J. -W. Kang, J. -S. Kim, C. -M. Lee, E. Kim, and J. -J. Kim, "1×2 all-optical switch using photochromic-doped waveguides," *Electronics Letters*, vol. 36, no. 19, pp. 1641–1643, September 2000.

[5] C. Kim, D. A. May-Arrioja, P. Newman, and J. Pamulapati, "Ultrafast all-optical multiple quantum well integrated optic switch," *Electronics Letters*, vol. 36, no. 23, pp. 1929–1930, November 2000.

[6] C. De Matos, M. Pugnet, and A. Le Corre, "Ultrafast coherent all-optical switching in quantum-well semiconductor microcavity," *Electronics Letters*, vol. 36, no. 1, pp. 93–94, January 2000.

[7] D. K. Hunter and D. G. Smith, "New architectures for optical TDM switching," *IEEE/OSA Journal of Lightwave Technology*, vol. 11, no. 3, pp. 495–511, March 1993.

[8] H. F. Jordan, D. Lee, K. Y. Lee, and S. V. Ramanan, "Serial array time slot interchangers and optical implementations," *IEEE Transactions on Computers*, vol. 43, no. 11, pp. 1309–1318, November 1994.

[9] J. Yates, J. Lacey, and D. Everitt, "Blocking in multiwavelength TDM networks," in *4th International Conference on Telecommunication Systems, Modeling, and Analysis*, March 1996, pp. 535–541.

[10] F. P. Kelly, "Blocking probabilities in large circuit switched switched networks," *Advances in Applied Probability*, vol. 18, pp. 473–505, May-June 1986.

[11] A. Girard, *Routing and dimensioning in circuit-switched networks*, Addision-Wesley, 1990.

[12] M. Kovacevic and S. Acampora, "On wavelength translation in all-optical networks," in *Proceedings of IEEE INFOCOM'95*, April 1995, pp. 413–422.

[13] A. Birman, "Computing approximate blocking probabilities for a class of all-optical networks," in *Proceedings of IEEE INFOCOM'95*, April 1995, pp. 651–658.

[14] E. Karasan and E. Ayanoglu, "Performance of WDM transport networks," *IEEE Journal of Selected Areas in Communications*, vol. 16, pp. 1081–1096, September 1996.

[15] M. Kovacevic and S. Acampora, "Benefits of wavelength translation in all-optical clear-channel networks," *IEEE Journal of Selected Areas in Communications*, vol. 14, no. 5, pp. 868–880, June 1996.

[16] R.A. Barry and P.A. Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers," *IEEE Journal of Selected Areas in Communications*, vol. 14, no. 5, pp. 858–867, June 1996.

[17] N. Wauters and P. Demeester, "Is wavelength conversion required?," in *IEEE International Conference on Communications'96 Workshop on WDM Management*, June 1996.

[18] S. Subramaniam, M. Azizoglu, and A.K. Somani, "All-optical networks with sparse-wavelength conversion," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 544–557, August 1996.

[19] L. Li and A.K. Somani, "A new analytical model for multi-fiber WDM networks," in *Proceedings of the Global Telecommunications Conference, GLOBECOM'99*, 1999, vol. 1B, pp. 1007–1011.

[20] N. Wauters and P. Demeester, "Wavelength conversion in optical multi-wavelength multi-fiber transport networks," *International Journal of Optoelectronics*, vol. 11, no. 1, pp. 53–70, Janurary/February 1997.

[21] R. Ramaswami and G. H. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," in *Proceedings of IEEE INFOCOM'97*, April 1997, pp. 489–498.

[22] R. Ramaswami and G. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 744–754, December 1998.

[23] J. Yates, J. Lacey, D. Everitt, and M. Summerfield, "Limited-range wavelength translation in all-optical networks," in *Proceedings of IEEE INFOCOM'96*, March 1996, vol. 3, pp. 954–961.

[24] V. Sharma and E. A. Varvarigos, "Limited wavelength translation in all-optical WDM mesh networks," in *Proceedings of IEEE INFOCOM'98*, March 1998, vol. 2, pp. 893–901.

[25] T. Tripathi and K.N. Sivarajan, "Computing approximate blocking probabilities in wavelength-routed all-optical networks with limited-range wavelength conversion," in *Proceedings of IEEE INFOCOM'99*, March 1999, vol. 1, pp. 329–336.

[26] A. Mokhtar and M. Azizoglu, "Adaptive wavelength routing all-optical networks," *IEEE Transactions on Networking*, vol. 6, no. 2, pp. 197–206, April 1998.

[27] S. Thiagarajan, "Traffic grooming and wavelength conversion in optical networks," *PhD Dissertation, Iowa State University*, Fall 2001.

[28] S. Subramaniam and R. A. Barry, "Wavelength assignment in fixed routing WDM networks," in *IEEE International Conference on Communications*, November 1997, pp. 406–410.

[29] H. Harai, M. Murata, and H. Miyahara, "Performance of alternate routing methods in all-optical switching networks," in *Proceedings of IEEE INFOCOM'97*, April 1997, pp. 516–524.

[30] E. D. Lowe and D. K. Hunter, "Performance of dynamic path optical networks," in *IEE-Proceedings of Optoelectronics*, August 1997, pp. 235–239.

[31] S. Ramamurthy and B. Mukherjee, "Fixed alternate routing and wavelength conversion in wavelength-routed optical networks," in *Proceedings of the Global Telecommunications Conference, GLOBECOM'98*, November 1998, pp. 2295–2303.

[32] L. Li and A. K. Somani, "Dynamic wavelength routing using congestion and neighborhood information," *IEEE Transactions on Networking*, vol. 7, no. 5, pp. 779–786, October 1999.

[33] H. Zang, L. Shasrabuddhe, J. P. Jue, S. Ramamurthy, and B. Mukherjee, "Connection management for wavelength-routed WDM networks," in *Global Telecommunications Conference, GLOBECOM'99*, 1999, vol. 2, pp. 1428–1432.

[34] J. Jue and G. Xiao, "An adaptive routing algorithm for wavelength-routed optical networks with a distributed control scheme," in *Proceedings of the Nineth International Conference on Computer Communications and Networks*, October 2000, pp. 192–197.

[35] R. Ramaswami and A. Segall, "Distributed network control for wavelength routed optical networks," *IEEE Transactions on Networking*, vol. 5, no. 6, pp. 936–943, December 1996.

[36] H. Zhang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, no. 1, pp. 47–60, January 2000.

[37] B. Wen and K. M. Sivalingam, "Routing, wavelength and time-slot assignment in time division multiplexed wavelength-routed optical WDM networks," in *Proceedings of IEEE INFOCOM'02*, June 2002.

[38] R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*, Addison-Wesley, 2nd edition, 1999.

[39] R. Srinivasan and A.K. Somani, "Dynamic routing in WDM grooming networks," *Technical Report (DCNL-ON-2001-001), Dependable Computing and Networking Laboratory, Department of Electrical and Computer Engineering, Iowa State University*, August 2001.

[40] L. H. Sahasrabuddhe and B. Mukherjee, "Light trees: optical multicasting for improved performance in wavelength routed networks," *IEEE Communications Magazine*, vol. 37, pp. 67–73, February 1999.

[41] Y. Yang, J. Wang, and C. Qiao, "Nonblocking WDM multicast switching networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, pp. 1274–1287, December 2000.

[42] Y. Wang and Y. Yang, "Multicasting in a class of multicast-capable WDM networks," in *Ninth International Conference on Computer Communications and Networks*, October 2000, pp. 184–191.

[43] X. Zhang, J.Y. Wei, and C. Qiao, "Constrained multicast routing in WDM networks with sparse light splitting," *Journal of Lightwave Technology*, vol. 18, no. 12, pp. 1917–1927, March 2001.

[44] N. Sreenath, K. Satheesh, G. Mohan, and C.S.R. Murthy, "Virtual source based multicast routing in WDM optical networks," in *IEEE International Conference on Networks (ICON 2000)*, September 2000, pp. 385–389.

[45] E. Iannone, M. Listanti, and R. Sabella, "Analysis of multicasting in photonic transport networks," in *International IFIP–IEEE Conference on Global Infrastructure for the information age*, April 1996, pp. 406–417.

[46] A. E. Kamal and A. Al-Yatama, "Blocking probabilities in circuit-switched WDM networks under multicast service," *Journal of Performance Evaluation*, vol. 47, no. 1, pp. 43–71, January 2002.

[47] S. Ramesh, G. N. Rouskas, and H. G. Perros, "Computing blocking probabilities in multicass wavelength-routing networks with multicast calls," *IEEE Journal of Selected Areas in Communications*, vol. 20, no. 1, pp. 89–96, January 2002.

[48] S. Subramaniam, M. Azizoglu, and A. K. Somani, "On the optimal converter placement in wavelength-routed networks," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 754–766, October 1999.

[49] S. Thiagarajan and A. K. Somani, "An efficient algorithm for optimal wavelength converter placement on wavelength-routed networks with arbitrary topologies," in *Proceedings of IEEE INFOCOM'99*, March 1999.

[50] M. Sridharan, "Design and operation of mesh-survivable WDM optical networks," *PhD Dissertation, Iowa State University*, Summer 2002.