IP MULTICAST OVER WDM NETWORKS

Ding Aijun
(M. Eng, NUAA)

A DISSERTATION SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2003

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my supervisor, Dr. Tan Sun Teck, and ex-supervisor, Dr. Poo Gee Swee. Dr. Tan is a friendly person. He has helped me a lot in both my research and my life. Dr. Poo continued supervising me unofficially after he left School of Computing, National University of Singapore. He provided me with valuable advice to guide me through the whole research course.

I am very glad to have many friends during my course of research. Some of them have already left Singapore. But the existence of each of them has created a warm family atmosphere. In particular, I would like to thank Jun He, Jinquan Dai and Zhengdong Yu for the constructive discussions on research matters. Additional thanks to Jun He for lending me the powerful PC for my simulations.

I feel deeply indebted to my wife. She is always with me through days and nights, happy or sad, in Singapore. The life here could be very tough for a young couple, especially at the time of economic recession. She has suffered a lot from the hard life.

Finally, I am very grateful to my parents. Unlike many Chinese parents, they never give me any pressure. Instead, they always give me enough freedom to choose my own way. And yet, they would correct me silently whenever I stray. They have instilled hope, confidence and perseverance, and that has helped me endure difficult times. Papa and Mama, you son has finally made it! I dedicate my thesis to you.

TABLE OF CONTENTS

List of Figures

LIST OF TABLES

SUMMARY

Multicast is a useful network service which aims to provide efficient communication services for applications that send data to multiple recipients. IP multicast is a receiver-based concept, and the sender does not need to maintain a list of receivers. Future multimedia group applications, such as video conferencing, interactive gaming, and network storage require huge bandwidth, and involve group communication. Multicast provides an efficient and economical way to employ bandwidth and network resources while the wavelength division multiplexing (WDM) technology provides huge bandwidth. The combination of multicast and WDM is critical for the future optical Internet.

Generally, there are three kinds of WDM networks, namely, wavelength-routed networks, broadcast-and-select networks, and packet-switched networks. Currently, a few algorithms have been developed to support multicast over each kind of networks. These algorithms are to be reviewed in this thesis.

We consider multicast over wavelength-routed networks. Previous works done in this area suffer a number of drawbacks. The most serious two are: 1) they assume infinite light splitting and wavelength conversion capabilities, and 2) they divide the problem into two sub-problems: multicast routing and wavelength assignment, and tackle them individually. The first drawback renders the algorithms useless to real WDM networks which have only finite capabilities. The second drawback results in inefficient routing results, as it does not consider routing and wavelength assignment as one unified problem.

To overcome the shortcomings, we develop an expanded-graph model to transform a WDM network into a graph, and then apply existing Graph Theory algorithms on the graph. Compared with other models, the expanded-graph model simplifies the

resulting graph significantly.

To further reduce computational complexity while maintaining a certain level of quality in the solution, we develop a layered-routing approach. The approach corresponds to the multi-layer nature of a WDM network. To facilitate algorithm development, we propose two generic algorithm frameworks. This allows any algorithm that is derived from either Prim or Kruskal to be easily imported to solve the problem.

Both methods can complete multicast routing and wavelength assignment in one phase. Network status is considered at the time of routing. Moreover, the methods are general enough for use in any kind of wavelength-routed networks. In particular, they can deal with those networks with finite light splitting and wavelength conversion. This is the first time that networks with finite capabilities are studied. The results should help network architects and engineers in designing an optical network.

We simulate all our proposed algorithms and measure their performance. In addition, we develop two analytical models – one for adaptive unicast routing and the other for fixed multicast routing. Both models are based on the queuing network theory, and their distinctive feature is that they take limited capabilities into consideration, which significantly increases the difficulty of analysis.

Finally, we also consider delay-constrained multicast over a WDM network. We anticipate the need for QoS routing in future in fast-speed WDM networks.

ACRONYM

ADH	Average Distance Heuristic
ARPANET	Advanced Research Projects Agency NETwork
ATM	Asynchronous Transfer Mode
DCP	Delay-constrained least-Cost Path
DCSP	Degree-Constrained Steiner Problem
DWDM	Dense Wavelength Division Multiplex
EDFA	Erbium-Doped Fiber Amplifier
GMPLS	Generalized MultiProtocol Label Switching
GSMP	Generalized Switch Management Protocol
MCRWA	MultiCast Routing and Wavelength Assignment
IP	Internet Protocol
LAN	Local Area Network
LCP	Least-Cost Path
LDP	Least-Delay Path
LP	Linear Programming
LS	Light Splitting
LSP	Label-Switched Path
MAC	Media Access Control
MAN	Metropolitan Area Network
MPLS	MultiProtocol Label Switching
MAW	Multicast with Any Wavelength
MSDW	Multicast with Same Destination Wavelength
MST	Minimum Spanning Tree
MSW	Multicast with Same Wavelength

OBS	Optical Burst Switching
OLS	Optical Label Switching
OXC	Optical CrossConnect
PSC	Passive Star Coupler
RAM	Random Access Memory
RPF	Reverse Path Forwarding
SMT	Steiner Minimum Tree
SPG	Steiner Problem in Graphs
SPH	Shortest Path Heuristic
VS	Virtual Source
WADM	Wavelength Add/Drop Multiplexer
WAN	Wide-Area Network
WC	Wavelength Conversion
WDM	Wavelength Division Multiplex
WIC	Wavelength InConvertible

Chapter 1 Introduction

1.1 Multicast in Various WDM Networks

The first laser was invented in 1958 [5] [43]. At that time, engineers and scientists were already aware of the tremendous potential of optical communication. However, the exploration in the early days was hard and slow. The breakthrough came in the early 1990s, when Erbium-Doped Fiber Amplifiers (EDFAs) became commercially available. For the first time in history, optical signals could be directly amplified in the optical domain. With that, the Wavelength Division Multiplexing (WDM) technology finally took off.

Following years of evolution, modern WDM networks may be classified as shown in Figure 1. The techniques to support multicast over each kind of network vary. We briefly touch on them here, and present them in detail in Chapter 2.

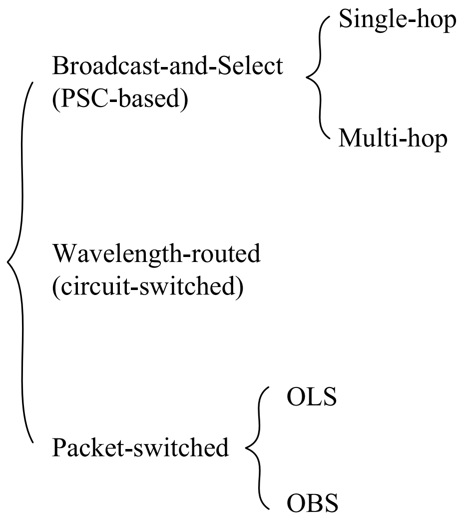


Figure 1 Taxonomy of WDM networks

The broadcast-and-select network centers on a shared medium, usually, a passive star coupler (PSC). Thus, it is sometimes called the PSC-based network. The physical topology of the broadcast-and-select network is usually a star, but it can also be a bus or a tree [16][17]. Broadcast-and-select networks have two logical topologies: single-hop or multi-hop, and are used mainly as LAN/MAN. The difference in logical topology differentiates single-hop networks from multi-hop ones.

Nodes in single-hop networks are normally equipped with tunable transceivers. A successful transmission requires fast tuning of either the transmitter or the receiver or both. Hence, the cost of agile transceivers and the scheduling involved is high. On the other hand, multi-hop networks aim to alleviate cost by using low-cost fixed transceivers. Multi-hop networks usually adopt regular logical topologies, such as ShuffleNet, Hypercube, or Manhattan Street Network [17]. A logical topology is almost static; it does not change unless the change is beneficial.

However, the single-hop structure makes more sense for multicast transmission because the star coupler is equipped with intrinsic multicast, or more precisely, broadcast capability. Therefore, almost all existing works deal with multicast over single-hop networks. The key is to design an efficient media access control protocol to schedule transmission and the tuning of transceivers [3][16].

Optical packet switching technology is still in its infancy. The key techniques to enable packet switching, such as optical RAM, and quantum computing, are far from being practical. Currently, a few interim substitutes have been proposed, such as Optical Label Switching (OLS) [35] and Optical Burst Switching (OBS) [58][59]. Basically, OLS uses the sub-carrier multiplex technique [92] to carry labels together with data packets so as to facilitate instant switching. On the other hand, OBS assembles data packets into larger bursts and then transfers each burst after a control packet, which reserves resources along the route for the burst. Multicast over OBS has been studied [58][59]. The problem of multicast over OLS can be solved in the framework of GMPLS.

However, all the above networks are out of our scope. Our target is to support multicast over wavelength-routed networks. These networks usually serve as WANs, and adopt the mesh topology. They rely on wavelength for routing. Therefore, the interim optical switches must know the relationship between the wavelength and the routing decision in advance. So a virtual circuit has to be established before real data transmission begins. In this sense, we view the wavelength-routed network as one kind of circuit-switched network.

Different from an electronic network where each physical cable can support only one channel, multiple channels can be supported in a single fiber in a WDM network. Each channel resides on one wavelength. A new problem – wavelength assignment – is raised, in addition to multicast routing. Therefore, multicast over WDM involves two sub-problems: routing and wavelength assignment. The problem is often called the multicast routing and wavelength assignment (MCRWA) problem. This is the topic of this thesis.

1.2 Motivation of Research

With the rapid increase in computational power and network bandwidth, applications have become more and more reliant on multimedia representation and group collaboration. For instance, a remote medical system, which brings doctors all over the world together, needs a huge bandwidth to transmit medical images. Most of the images will be uncompressed and huge in volume, as compression may damage them. Such multimedia group applications and other future applications require huge bandwidth, and an efficient method of utilizing bandwidth needs to be developed. Multicast is a good and natural choice.

To enjoy the benefits of WDM multicasting [90], light splitting and wavelength conversion capabilities must be fully deployed. Both capabilities are unique to the WDM network. By using light splitting, the time-consuming and expensive packet copying procedure in the electrical domain is substituted by quick and cheap light splitting in the optical domain. By using wavelength conversion, the blocked path can be detoured to a different wavelength. Hence, blocking performance is improved.

Wavelength-routed networks can be classified using various criteria. In terms of light splitting capability, they can be classified into two groups: one with full light splitting, and another with finite light splitting. The criterion is the percentage of the nodes that have light splitting capability. Similarly, networks can also be grouped according to their wavelength conversion capability.

For multicast over full splitting networks, the problem is divided into two sub-problems: multicast routing and wavelength assignment. The multicast routing sub-problem is usually tackled by using Steiner Minimum Tree (SMT) heuristics [29] while the wavelength assignment sub-problem is generally modeled as a graph-coloring problem [36]. Usually, wavelength continuity constraint is imposed onto wavelength assignment; that is, no wavelength conversion is allowed anywhere.

In general, there are three approaches to multicast over sparse splitting networks. These approaches will be elaborated in Chapter 2. These approaches suffer from a few problems, and two of them are serious and of general concern.

First, the approaches usually divide the problem into two parts: multicast routing and wavelength assignment, and then tackle them individually. The two-phase scheme is not efficient as the routing decision is made without considering network utilization status. Therefore, a multicast tree may not be successfully embedded into the network, especially when the network is heavily loaded.

Second, the approaches only consider the networks with infinite light splitting and wavelength conversion capabilities. However, due to various restrictions such as cost, technique and so on, WDM networks only have finite capabilities in real situations.

Ideally, our proposal should overcome as many problems as possible while maintaining a high level of performance. Our basic idea is to transform WDM networks into appropriate graphs, and then apply Graph Theory algorithms to solve the problem. The graph transformation method, together with Delay-Constrained Steiner Problem (DCSP) heuristics (cf. Section 4.3.4), helps complete multicast routing and wavelength assignment in one phase, and successfully address the finite nature of WDM networks. We develop and propose a few approaches [2][4], and will discuss them in detail later.

In addition, we attempt to solve delay-constrained multicast over WDM networks. With the development of network applications, we anticipate the need for such QoS multicast in the future.

1.3 Objective of Research

In a broad sense, multicast is a kind of group communication, either one-to-many, or many-to-many, or many-to-one. In this thesis, we consider multicast as one-to-many communication, i.e., selective broadcast. Typical IP multicast has three parts: multicast addressing, group membership management, and multicast routing. Our focus is on the multicast routing problem. Moreover, we will only consider multicast over wavelength-routed WAN. The final receivers are, of course, in their relevant LAN. Finally, once the signal enters the WAN, it remains in the optical domain before it finally exits the network. That is to say, all-optical transmission is required.

Multicast over WDM networks has benefited much from the light splitting and wavelength conversion capabilities of the WDM network. Hence, our transformation method must be able to produce graphs which correctly represent both capabilities, especially when the capabilities are finite. Finiteness has different meanings at different levels. At the network level, it means that not all the nodes have light splitting and/or wavelength conversion capability. At the node level, a node may not have an infinite number of light splitters and/or wavelength converters. At the component level, a light splitter or a wavelength converter has only finite capability. Finiteness at the first two levels is easy to deal with. To tackle finiteness at the component level, we introduce the degree-constrained Steiner problem.

Generally, Graph Theory algorithms are sensitive to the number of nodes and edges in a graph. Computational overhead increases fast with the number. Algorithms that operate on the whole transformed graph can generate high-quality solutions, however, at the cost of high computational complexity. Hence, we need to create another approach that runs faster by sacrificing a little in the quality of the solution. The approach is expected to make use of the multi-layer nature of the WDM network.

Routing approaches can be divided into three categories: static, alternate and dynamic [7] [36]. We prefer the dynamic routing approach. It outperforms other approaches because the routing decision is made after taking network utilization into consideration.

In conclusion, the objective of our research is to support offline multicast over WDM networks with finite light splitting and wavelength conversion capabilities by using the dynamic routing approach. We seek to do the following in our study:

- Complete multicast routing and wavelength assignment in one single step.
- Deal with situations where the network has finite light splitting and wavelength conversion capabilities.
- Create an effective method for transforming any WDM network into one single

graph. Then modify Graph Theory algorithms and apply them on the resultant graph.

- Find an alternate fast approach which operates on partial network once. In other words, only part of the graph is used for routing purpose. Hopefully, the quality of the solution will not degrade significantly.
- Test the performance of all algorithms by means of simulation. Conduct theoretical analysis of simple cases.
- Optionally, keep the proposed algorithms general enough for any wavelength-routed networks.

1.4 Contributions of Thesis

In the remainder of this thesis, we will investigate the problem of multicast over WDM networks. In total, we will develop the expanded-graph model and the layered-routing approach. Under the model, any WDM network can be transformed into a graph, and then the relevant Graph Theory algorithms can be applied to it with minor modifications. By the layered-routing approach, we will reduce computational complexity by sacrificing some quality of the solution. Using the approach, we will design two generic algorithm frameworks to provide guidance for modifying existing algorithms and applying them to solve the MCRWA problem. We will also conduct intensive simulations to test the performances. The simulation results will reveal the relationship among various parameters, such as the number of wavelengths, light splitting capability, and so on, and can be used to guide the design of a real optical network. Our contributions are listed below in detail.

- We develop the expanded-graph model, which is able to transform a WDM network into a graph. The transformation method is general, provided the

WDM network adopts the share-per-node scheme [50]. Under the scheme, the splitters and converters of one node are shared among all the connections at that node. The model is capable of completing multicast routing and wavelength assignment simultaneously. The resultant graph of our model is much simpler than that generated by the auxiliary-graph model [50].

- DCSP heuristics must be modified slightly before being applied to the graph generated by our model. We modify the Shortest Path Heuristic (SPH) [29] as an example. We test it on *real* WDM networks for blocking performance. By real, we mean that the network has only finite splitting and conversion capabilities. The simulation results show the blocking performance of optical networks under various constraints.
- The algorithms on the transformed graph are of high computational complexity because the graph has a large numbers of nodes and edges, and the DCSP heuristics are sensitive to those numbers. Therefore, we develop a layered-routing approach by utilizing the multi-layer nature of the WDM network. Computational complexity is reduced at the cost of marginal degradation of the quality of the solution. The approach completes routing and wavelength assignment in one phase.
- Using the layered-routing approach, we develop two generic algorithm frameworks to facilitate the use of graph-theory algorithms. The frameworks are the quasi-Prim framework and the quasi-Kruskal framework. They are capable of importing, respectively, the Prim and Kruskal algorithm families into the approach. In particular, we have already modified the following algorithms: naïve, SPH and its variants, K-SPH, and Average Distance Heuristic (ADH). The prototypes of the algorithms can be found in

[29][66][78]. The simulations are conducted over real WDM network to reveal the relationship between blocking performance and various characteristics of the network.

- Adaptive unicast routing is analyzed theoretically using queuing network theory. As WDM networks have only limited wavelength conversion capability, blocking happens not only at the links as in all other works encountered, but also at nodes. We propose an iterative algorithm to analyze overall blocking performance.
- Fixed multicast routing is analyzed theoretically as well. Compared with unicast, limited light splitting capability is added in. Here, on-tree nodes are modeled using several complex queuing networks. Blocking performance is deduced using our iterative algorithm.
- We conduct the first survey of multicast over WDM networks. The survey covers multicast over most popular network types: single-hop, and wavelength-routed. Various proposals on routing and wavelength assignment are investigated. The properties of multicast-capable switches are studied as well.
- To the best of our knowledge, we are the first to study the MCRWA problem over WDM networks with *finite* capabilities. By studying the practical WDM network, we seek to provide valuable guidelines for network design and configuration.
- We study delay-constrained multicast over dense WDM networks. A graph transformation method, which is based on the concept of the WIC region, is proposed. By transforming WDM networks to flat graphs using our method, abundant existing algorithms can then be used after minor modifications.

1.5 Organization of Thesis

The rest of this thesis is organized as follows. Chapter 2 presents a detailed survey of multicasting over various WDM networks, and investigates the pros and cons of various proposals. Chapter 3 introduces the simulation methodology that is applicable to all simulations conducted in the thesis. Chapter 4 introduces the expanded-graph model. The modified SPH algorithm is simulated and the results are presented as well. Section 4.2 provides the definitions, notations and formulations used in the thesis. Chapter 5 presents the simplified layered-routing approach. Two generic algorithm frameworks are presented, together with the algorithms developed under the frameworks. The approach is compared with the model, and guidelines are given to help make a choice. Chapter 6 presents the analytical models for analyzing blocking performance in WDM networks. Chapter 7 touches on delay-constrained routing over WDM networks. Chapter 8 concludes the thesis, and indicates the directions for future work. Appendix A includes brief technical issues on WDM and IP over WDM.

Chapter 2 Review of Related Work

2.1 Introduction

The concept of multicast has been widely studied in traditional packet-switched networks. Multicast applications, such as multimedia conferencing, medical imaging, and digital audio demand huge bandwidth support. Advancements in WDM technology have provided the availability of enormous bandwidth (about 30THz [16]). It is natural to extend the multicast concept to optical networks in order to gain enhanced performance.

Multicast can be supported at many layers. First, at the hardware layer, multicast capability can be built inside optical switches and other devices. Second, multicast can be supported at the WDM layer to take advantage of the light splitting capability of optical switches. Third, multicast can be run at the IP layer. If all optical routers (switches) understand IP protocol, then most of the IP multicast protocols can be applied without modification. However, this scheme does not take the special characteristics of WDM networks into account. So it may be low in efficiency. Finally, multicast can be supported at higher layers, such as in some reliable multicast protocols.

The final target of optical network evolution will be pure packet switching. However, restricted by technology, the wavelength-routed network, which is similar to the circuit-switched network, forms the majority. Multicasting over wavelength-routed networks requires the reservation of a wavelength at each branch of the multicast tree (or forest). It is suitable for high bandwidth multicast applications with relatively long duration. Wavelength-routed networks usually adopt the mesh topology, and serve as WAN.

On the other hand, there are several interim optical packet switching techniques, such as Optical Label Switching (OLS) [35] and Optical Burst Switching (OBS) [58] [59]. Multicast over such networks raises different requirements. Take OBS as an example. Multicast over such networks requires every switch to maintain a WDM forwarding cache. Switching is based on fixed-sized labels that are carried using the subcarrier multiplexing technique [35][92], or other suitable techniques.

Besides packet-switched networks, WDM networks can be grouped into two categories: broadcast-and-select networks (mostly used in LAN/MAN) and wide-area mesh networks. Broadcast-and-select networks may be divided into two sub-categories: single-hop and multi-hop. However, few articles address the problem of multicast over multi-hop networks using passive star couplers. Therefore, we investigate two types of networks: single-hop and wavelength-routed. Supporting multicast in each category faces distinctive problems. The solutions provided are different too.

In both categories of networks, the key components of multicast are light splitters and wavelength converters. The principles and operations of these components are described in Appendix A. WDM Technology. In single-hop networks, the hub is usually a passive star coupler, a kind of splitter. In mesh networks, the building block for a light-tree [52] is definitely a light splitter.

Multicasting over a single-hop network is relatively easy. This network usually adopts a star topology, with a passive optical splitter at the hub and each station connects directly to it. Each station has a number of transmitters and receivers, either tunable or fixed. Before a successful transmission, the pair of transmitter and receiver should tune to the same wavelength. Therefore, the key point here is to find an efficient scheduling algorithm to coordinate the tuning. Many scheduling algorithms exist, which can be classified into three categories, namely, random-access based [28], reservation based [94] and pre-allocation based [34] scheduling algorithms.

Multicasting over a wide-area network is rather complex. It needs to consider issues of IP over the optical network. In a wavelength-routed network, a lightpath is used for unicast communication whereas a light-tree is used for multicast. A lightpath can be established statically or dynamically [40] [41]. So can a light-tree.

By and large, wide area optical multicast can be classified into two groups in terms of light-splitting capabilities: one with full light splitting, and another with sparse light splitting. A light-tree [52] makes use of the light splitting capability of nodes and provides a means of transportation for multicast traffic over the all-optical network. Using light splitting, the time-consuming and expensive packet copying procedure in the electrical domain is substituted by quick and cheap light splitting in the optical domain. Thus, WDM multicast [90] achieves better performance.

In sparse light splitting networks, however, some nodes may not have light splitting capability. It is then advisable to avoid using these nodes because they require either packet copying or wavelength conversion, which is costly. As a result, a single multicast tree may be inadequate for multicasting data to all destinations. In this case, several light-trees rooted at the same source node are to be used to form a multicast forest [90].

Several multicast routing protocols have been proposed to support multicast over WDM mesh networks. These protocols attempt to use light splitting capability as much as possible. The protocols can be categorized into three major groups. The first group aims to map the existing multicast tree built at the IP layer onto the WDM layer [24][89][90]. The second group builds a multicast forest in a centralized way with the full knowledge of the network [90]. The third group attempts to build the multicast tree around powerful nodes, such as virtual sources [61][62][83].

In the rest of this chapter, we provide a comprehensive review of the main concept related to optical multicast, optical switches with their splitting capabilities, multicast over single-hop, and wavelength-routed networks. We also discuss the related challenging algorithms developed for multicast routing in the optical domain.

2.2 Multicast-Capable Optical Switches

Optical switches which possess light splitting capability are said to be multicast capable (MC). So, an MC switch needs to use splitters or passive star couplers as components [37][72]. These switches are capable of multicasting data in the optical domain. In IP networks, a multicast incapable (MI) router is a router that does not understand or run the appropriate IP multicast protocol. However, in WDM networks, an MI switch means it has no light splitting capability.

2.2.1 Splitter and Delivery Switch

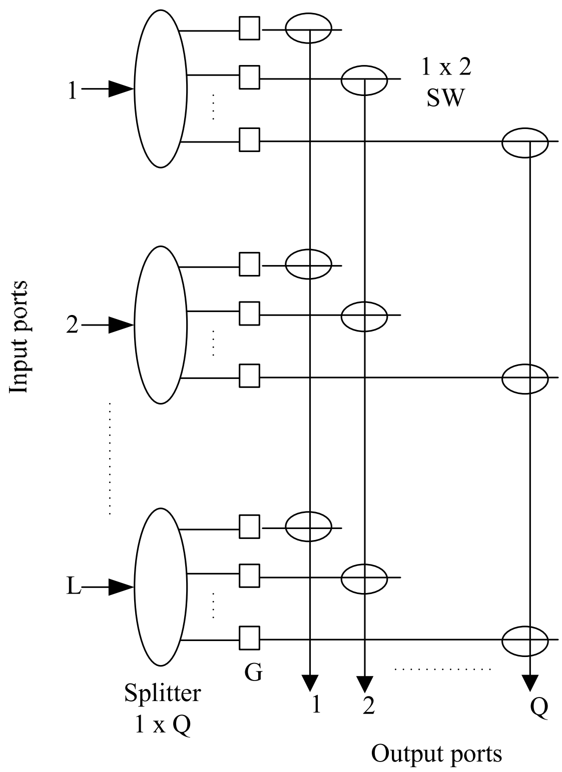


Figure 2 A proposed L x Q SAD switch by using 1 x Q splitters, optical gates (G) and 1 x 2 switches

A number of MC switch architectures have been proposed. For example, a conceptual MC switch architecture based on linear combiners and dividers is presented in [52]. A so-called splitter-and-delivery (SAD) switch (see Figure 2) is proposed in [82]. The optical gates are optional in the structure, but used to reduce crosstalk. This device has a high modularity. It is suitable for planar silica waveguide technology [37][72]. All these components can be integrated on a single silicon chip.

The SAD switch works as follows. A beam of light is split into Q branches. Each branch is connected to one of Q tiny *1* x *2* switches. Each tiny switch can either switch the branch to a fixed output port or discard it. Therefore, an input light can be switched to *none*, *one*, *more*, or *all* output ports. Hence, the SAD switch is a strictly nonblocking switch. Besides, multicasting can be done easily by selectively passing or blocking corresponding branches.

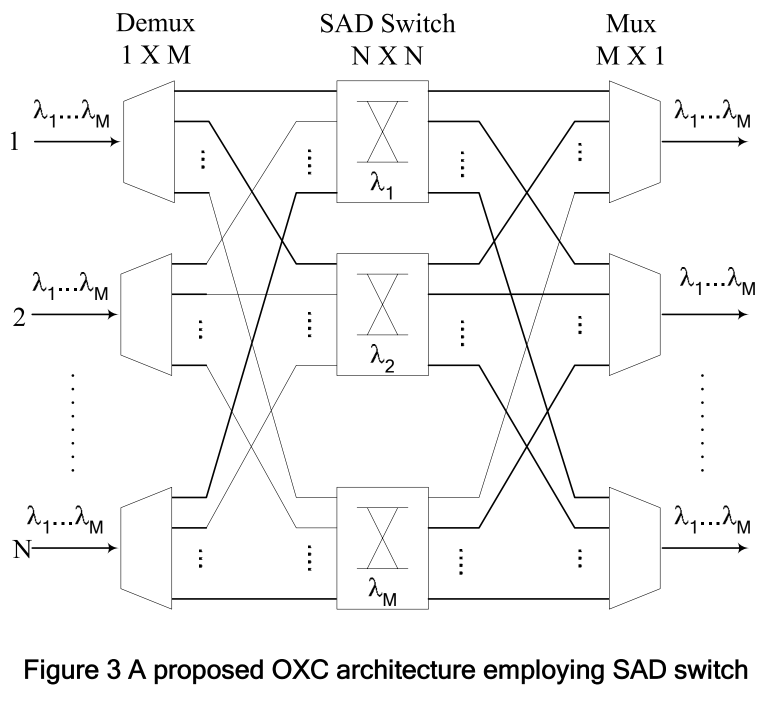


Figure 3 A proposed OXC architecture employing SAD switch

In real situations, the SAD switch is used as a stage of a bigger OXC (Optical Crossconnect). For instance, in Figure 3, a SAD switch is used to deal with a designated wavelength. Hence, the OXC has multicast ability.

2.2.2 Capacity and Cost Estimate of Nonblocking Multicast Switches

The properties of nonblocking WDM multicast switches are discussed in [93]. The properties include network capacity, cost, and nonblocking conditions, etc. The work sheds light on the design of multicast switches. For instance, the WP-OXC constructed by SAD switches proposed in [82] may be analyzed under the Multicast with Same Wavelength (MSW) model.

A *multicast connection* includes one source port and a few destination ports. According to the wavelength assignment to the ports, there are three different *multicast models*, namely, the Multicast with Same Wavelength (MSW) model, the Multicast with Same Destination Wavelength (MSDW) model and the Multicast with Any Wavelength (MAW) model (see Figure 4). A *multicast assignment* is a set of multicast connections that do not involve the same source node and the same destination node. Then, the *multicast capacity* of a WDM network (switching fabric) is defined as the number of multicast assignments that can be accommodated.

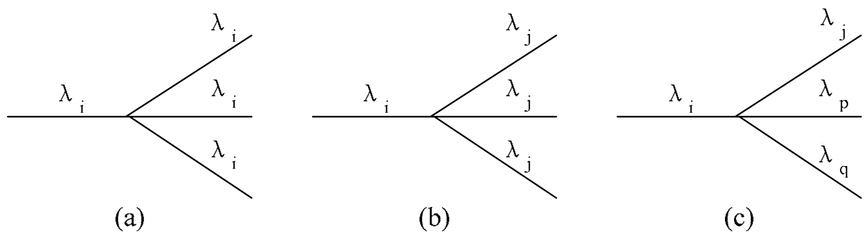


Figure 4 Three multicast models (a) MSW (b) MSDW (c) MAW

The network (switching fabric) is constructed with light splitters, wavelength converters, SOA gates and other optical components in the crossbar (single-stage) or multistage structure. *Network cost* is defined in terms of the number of crosspoints, which are actually the SOA gates or wavelength converters used.

Capacity of crossbar networks, nonblocking conditions of multi-stage networks, and cost of both networks are presented in [93]. The results concerning network cost are presented in Table 1. In the table, **N** equals the number of input/output ports, while k is the number of wavelengths each port has. The WP-OXC shown in Figure 3 is comparable to the MSW/CB entry in the table.

We may conclude that the MSDW model is not cost-effective in either crossbar or multistage networks, while the MSW and MAW models represent cost-performance trade-offs in the design of both crossbars and multistage networks.

Table 1 Cost comparison of multistage and crossbar WDM multicast networks under different models (CB-Crossbar, MS-Multistage)

Model	#Crosspoints	#Converters
MSW/CB	$k\mathrm{N}^2$	0
MSW/MS	$O(kN^{\frac{1}{2}}\frac{\log N}{\log\log N})$	0
MSDW/CB	$k^2\mathrm{N}^2$	$k\mathrm{N}$
MSDW/MS	$O(k^2N^{\frac{1}{2}}\frac{\log N}{\log\log N})$	$O(kN\frac{\log N}{\log\log N})$
MAW/CB	$k^2\mathrm{N}^2$	$k\mathrm{N}$
MAW/MS	$O(k^2N^{\frac{1}{2}}\frac{\log N}{\log\log N})$	$k\mathrm{N}$

2.3 Multicast in Single-Hop WDM Networks

2.3.1 Network Structure

Single-hop networks are also known as broadcast-and-select networks [16][28][34][39][84][94]. They are mainly used in local and metropolitan area networks. Most single-hop networks adopt the star topology for its simplicity, reliability and robustness (see Figure 5). Single-hop networks transmit messages directly from source to destination.

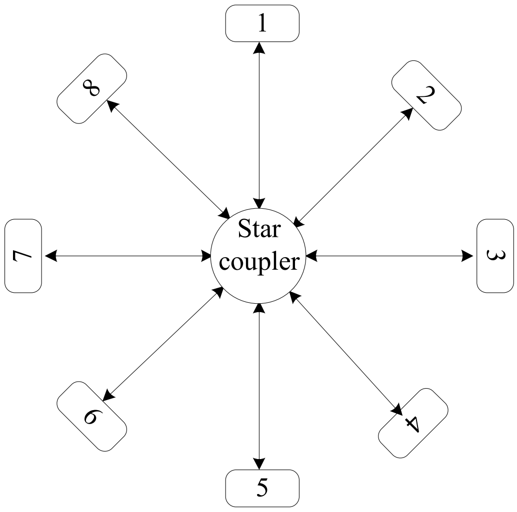


Figure 5 Star topology

A passive star coupler serves as a hub. A message is broadcasted through the star coupler. Each receiver selects the messages needed by tuning to the same wavelength as the sender. No matter how the technique varies, the distance from the source to the destination remains one hop. As the star coupler is a multi-access medium, a media access control (MAC) protocol is needed.

A network model is usually proposed to describe the protocol design. The system parameters include the number of nodes, the number of wavelengths, and the number and type (fixed or tunable) of transceivers each node is equipped with. Usually, there is a wavelength dedicated for control, i.e., the control channel. A centralized scheduler may be provided. Alternatively, a more popular approach is to make each node run an identical scheduling algorithm to provide synchronized decisions. Most recent systems adopt the time-division multiplexing scheme for data/control channels.

In the development of scheduling protocols, some key points have to be considered carefully. First, three kinds of conflicts may occur, namely, data channel conflict, control channel (if any) conflict and receiver conflict. The protocol should avoid any conflict as far as possible, and deal with the situation should any conflict really occur. Second, the tuning latencies of transceivers, though neglected in most protocols, do heavily affect the design of the protocol and its performance. Finally, propagation latencies from one node to another are also very important in protocol design. In practice, most protocols assume equal distance from the nodes to the star coupler. Therefore, identical propagation latency makes the design easier.

2.3.2 Scheduling Algorithm

A scheduling protocol is needed to handle concurrent multiple access. Many scheduling algorithms for coordinating concurrent data transmissions have been proposed in the literature. These algorithms can be classified into three categories, namely, random-access based [28], reservation based [94] and pre-allocation based [34] scheduling algorithms.

The random-access scheduling algorithm presented in [28] is applied in a LAN environment, which consists of about 100 nodes connected to a broadcast star hub with 32 wavelengths. Each node has one fast tunable transmitter (TT) and one fast tunable receiver (TR), operating with a dedicated control channel (CC), giving rise to a CC-TT-TR configuration. Unlike other networks, a simple master/slave scheduler runs in the hub and the node respectively. The scheduling algorithm runs on the master scheduler. In a network operating at 10 Gbps per channel, millions of schedules will be generated every second. Therefore, the scheduling algorithm should be simple, not sophisticated. Whenever a channel is available, the scheduler will randomly pick a waiting node to send data. The message will be sent continuously until all receivers have received it successfully. There are two retransmission schemes, persistent or backoff. Simulation results show that backoff retransmission performs better.

A reservation-based multicast protocol based on the concept of a virtual receiver is proposed in [94]. The network adopts the FT-TR model, i.e., each node is equipped with one fixed transmitter (FT) and one tunable receiver (TR). The transmitter requires a dedicated home channel, which ensures that the network is free of data channel conflict. Nonetheless, this requirement makes the system non-scalable. Unlike many other algorithms, which assume infinite tuning speed and omit tuning delay, the study in [94] considered tuning latency. Based on the concept of a virtual receiver – a set of physical receivers that behave identically in terms of tuning, the multicast-scheduling problem is transformed into a unicast one. Hence, any unicast scheduling algorithm can be adopted. Unfortunately, partitioning physical receivers into a set of virtual receivers is NP-complete difficult (except in two extreme cases: all-to-all broadcast and disjoint multicast groups). So, four heuristics, namely greedy-join, random join, greedy split and random split, were proposed to address the problem.

A suite of adaptive multicast protocols which belong to the pre-allocation type were developed in [34]. The system adopts the FT-TR model. Although FT is used, unlike in [94], no home channel is reserved for each node. Hence, data channel conflicts do occur. Each node maintains separate queues for packets destined for different nodes, and another separate queue for all received multicast packets. That is, the number of queues that each node maintains equals the number of nodes in the network. All channels are slotted, with each slot time equivalent to packet transmission delay plus tuning latency.

According to the average duration of the multicast session and the average size of a multicast group, multicast traffic may be divided into three types as follows:

Short Session and Small Group: treat multicast as multiple unicast (type I)

Short Session and Large Group: treat multicast as broadcast (type II)

Long Session: treat multicast as multicast (type III)

Any unicast scheduling algorithm can be used to transmit type I traffic. For type II traffic, the scheduling is simple. First, unicast and multicast (actually, broadcast) traffic are scheduled separately. Then the two schedules are merged using the proposed Schedule-Merging Heuristic (SMH). However, type III traffic is the hardest to deal with.

To accommodate type III traffic, all the slots are divided into two categories: synchronization slot and free slot. Between two synchronization slots, there are a number of consecutive free slots. A multicast source claims the owner of the consecutive free slots during the synchronization slot, and transmits. A unicast schedule is made in advance, and is changed when there is multicast traffic. By doing so, data channel conflict is avoided. However, each node must be aware of the group membership in order to improve channel utilization and avoid receiver conflict. To maintain membership information, two protocols are proposed, namely the Global-knowledge Multicast Protocol (GMP) and the Control-Packet Multicast Protocol (CMP).

A number of comparisons have been made on the performance and other merits of the proposed protocols. However, there is no consensus on which protocol is the best. The sensible way is to adopt a multicast mechanism that suits the network model, application type, and traffic characteristics.

2.4 All-Optical Multicast over Wide-Area WDM Network

This section discusses all-optical multicast. If O/E/O conversion is allowed, the problem will be similar to multicast over traditional networks. One way to classify the approaches is presented in [47]. In this case, two different contexts exist – one attempts to migrate IP multicast protocols onto optical label switching networks while the other focuses on the configuration of the WDM layer as part of the multicast routing and wavelength assignment (MCRWA) problem. This classification, though sensible, does not cover all cases of interest.

We adopt a different taxonomy by grouping the protocols according to their assumption on the density of light splitting capability (be it full or sparse). This is because most protocols separate multicast routing and wavelength assignment into two steps. Under the all-optical constraint, and given the fact that there is no optical computing capability and limited optical buffer, the only feasible way is to use a light-tree as the underlying transmission mechanism.

2.4.1 Multicast Routing over Full Splitting Networks

For full light splitting networks, all nodes are assumed to have ample light splitting capability. A lightpath is an optical path, which crosses several nodes and possesses wavelength continuity property. Owing to the wavelength continuity property, transmission between lightpath endpoints requires no processing or buffering at intermediate nodes. The property avoids abusing the wavelength conversion ability of the nodes and improves the whole network's performance.

Light-tree is a point-to-multipoint extension to lightpath aiming to provide an underlying infrastructure for multicast in an optical network [52]. The light-tree scheme uses extensively the light splitting capability of each node. It assumes that all nodes have adequate light splitting capability, which makes up the full light splitting network. At each branching point, the beam of laser light will be split into a certain number of sub-beams, which turns a lightpath into a light-tree.

Most MCRWA algorithms assume wavelength continuity constraint, which makes wavelength conversion redundant. These algorithms are suitable for WDM networks with full light splitting capability. They can be classified into two types: static and dynamic. The static algorithms can be sub-divided into fixed and alternate types [7][36].

Usually, some Steiner Minimum Tree (SMT) heuristics [29] or Linear Programming (LP) techniques are used to calculate the spanning tree over a multicast group. In static MCRWA, the trees for any multicast group are determined in advance, without taking link utilization into consideration. In other words, to a multicast group, the network is flat, independent of the number of wavelengths and the traffic load. If only one SMT tree is prepared for a multicast group, it is a fixed approach. However, if two or more SMT trees are prepared for a multicast group, it is called an alternate approach. A search algorithm is used to pick a suitable wavelength to hold the SMT tree, giving rise to wavelength assignment. The alternate approach has more chance to succeed than the fixed one.

In contrast, in dynamic MCRWA, link utilization on each wavelength is recorded. Thus, the topologies on each wavelength are different. One can run the SMT or LP algorithm on each wavelength and choose the optimum one.

In comparison, static MCRWA is relatively simpler than dynamic MCRWA. An example of static MCRWA is given in [36]. However, dynamic MCRWA achieves higher performance. In addition, it is possible to impose delay-bound constraint on the MCRWA problem to get interesting results [88].

2.4.2 Multicast Routing over Sparse Splitting Networks

In sparse light splitting networks, not all nodes have light splitting capability. Instead, one needs to make full use of the nodes with light splitting capability and avoid those without as light splitting capability is essential for WDM multicast.

By and large, there are two groups of protocols. One group aims to map the multicast tree built at the IP layer onto the WDM layer. Protocols belonging to this group require the collaboration of IP multicast routing protocols. Another group builds the multicast tree from scratch at the WDM layer. Some protocols belonging to this group require global knowledge of the network topology. Some are based on the virtual source concept. Others develop their own schemes. As a result of sparse light splitting, a single multicast tree may be insufficient for multicasting data to all the destinations. In this case, a multicast forest is to be formed.

Protocols based on modification to an existing multicast tree

First, a multicast tree is built at the IP layer using the IP multicast routing protocol. This is then mapped onto the WDM multicast tree using the WDM multicast protocol. The mapping will most likely result in a multicast forest, owing to the fact that some nodes are multicast incapable (MI).

Though not explicitly stated in papers [24][89], a trigger is really needed to start the mapping. In multicast over MPLS protocols, a similar mapping exists. Three trigger modes for MPLS have been proposed, namely, traffic driven (or flow driven), topology driven, and request driven (or control driven/session driven). Similar trigger modes can be applied to WDM networks. In addition, IP routers and optical switches lie on different layers. A protocol is needed for them to collaborate. Use of the Generalized Switch Management Protocol (GSMP) is suggested [9][10][57].

Someone-initiated LSP setup

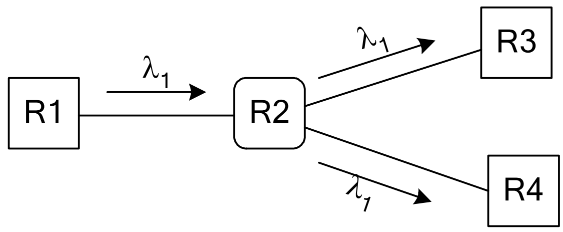


Figure 6 An example multicast tree

To facilitate explanation, a sample multicast tree, formed by an IP multicast protocol, is given in Figure 6. Under normal circumstances, if R2 has enough light splitting capability, then the WDM multicast tree will be exactly as shown in Figure 6. A problem arises if R2 is an MI node or R2 already exceeds its light splitting capability – in short, when R2 can only support one downstream node, say R3. A new LSP would then be needed to connect R4 back to the multicast tree.

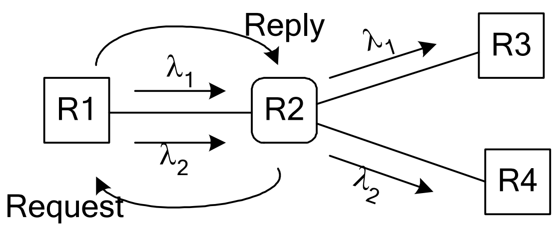


Figure 7 Parent-initiated LSP setup

A node can be designated as a parent, a child or a relative [89]. The relationship between nodes is decided by the Reverse Path Forwarding (RPF) rule. A parent-initiated LSP setup is illustrated in Figure 7. The optical switch at R2, which is the parent of node R4, can send a REQUEST to R1 to ask for another LSP. If R1 can afford another LSP, it will send back a REPLY, setting up a new LSP. In Figure 7, another wavelength λ_2 is consumed. λ_2 is determined by the wavelength assignment algorithm. Another possible way is to use only one wavelength between R1 and R2. In this case, R1 has to make two copies and sends them sequentially. Here, it is assumed that R1 has enough optical buffer for use.

Repair, purge and grow protocol

This protocol is proposed in [24]. It uses two messages, REPAIR and PURGE and one mechanism (i.e., grow back) to set up a multicast forest in the WDM layer.

After building up the multicast tree, a procedure is used to map the tree down to the WDM layer. The procedure starts with the sender sending out a REPAIR message. All the nodes, including the root, are categorized into two types: multicast capable (MC) or multicast incapable (MI). The number of REPAIR messages that a node can forward depends on which type it is, after receiving the REPAIR message. If it is an MC node, then it simply forwards the REPAIR message to all its downstream nodes. Otherwise, it can only pick one of its downstream nodes to forward the REPAIR message to, and has to send all the other nodes a PURGE message.

When a node receives a REPAIR message, it acts in the way described above. However, when a node receives a PURGE message, it has to grow back. The grow-back procedure is similar to the someone-initiated protocol described earlier. The main point is to set up a new LSP (lightpath). After the purged node successfully grows back, it can continue forwarding the repair message according to the above rule. The procedure ends successfully after all nodes of the IP multicast tree have received the REPAIR message. At that time, a multicast forest is also constructed.

Re-route-to-source and re-route-to-any

These two protocols are described in [90]. Each node in the network has two values attached: *splitting degree* and *fan-out degree*. Splitting degree decides the maximum number of downstream nodes that can be supported using light splitting capability. For example, splitting degree equals *1* for an MI node. The fan-out degree of a node equals the number of downstream nodes that are attached to it. It is assumed that the source has ample light splitting capability.

After building up the IP multicast tree, a procedure begins checking every node on the tree. The purpose is to see whether the node exceeds its light splitting capability or not, by comparing the splitting degree and the fan-out degree. If the fan-out degree is greater than the splitting degree, it means that the node has exceeded its splitting capability. Then, the number of outgoing branches with light splitting will be smaller.

Each of the affected child nodes can grow back to the tree (forest) in two ways, either Re-route-to-Source or Re-route-to-Any. The former method requires finding and joining one MC node along the reverse shortest path to the source. The latter method only needs to find one MC or leaf-MI node along any path on the tree.

Protocols requiring full knowledge of network

Two protocols, namely member-first, and member-only, are proposed in [90]. They require full knowledge of the whole network, such as the distribution of the MC switches. A heuristic tree formation algorithm is used to construct a multicast tree that avoids branching at the MI switches. The multicast forest is calculated in a centralized way, just like the MOSPF protocol. Actually, the problem of finding a multicast tree in a network where the nodes have limited light splitting capability is similar to the Degree-Constrained Steiner tree Problem (DCSP) [76].

As mentioned previously, each node holds two parameters: splitting degree and fan-out degree. The source is assumed to have ample light splitting capability. The links are all bi-directional in the network.

In the member-only algorithm, a multicast tree is constructed by including one member at a time (the closest member first). This algorithm is very similar to the naïve heuristic [57]. The member-first algorithm considers both membership information and distance among members when constructing a forest. In addition, it avoids branching at MI nodes. The multicast forest is constructed one tree at a time, and each tree is constructed link-by-link using a quasi-Dijkstra's algorithm. Actually, the member-first algorithm is much similar to Prim's algorithm for the Minimum Spanning Tree. The candidate links are organized in a priority queue, where a link leading to a member has a higher priority. Tree adjustment is carried out after a link is added, in order to expand the tree only on the MC or leaf-MI node. If this rule is violated, then the affected nodes and links have to be detached from the multicast tree and wait for future expansion. When all the members are included, the algorithm stops spanning the tree and starts pruning those branches that do not lead to any member.

Protocols that use powerful nodes as cores

These protocols try to expand a multicast forest around powerful nodes [61][62][83]. Powerful nodes are at least MC nodes. The protocols are similar to those described above to some extent.

The protocol as described in [83], first divides the members of a multicast group into the multicast-capable group (MCG) and the multicast incapable group (MIG). Then, a multicast tree is constructed for MCG. Next, MIG is further divided into sub-groups according to the distances from the MCG members. Finally, sub-trees are constructed among MCG members and its associated sub-MIG.

In [61] and [62], the nodes are further subdivided into four types: *Drop and Continue node* (*DaC-node*), *wavelength conversion node* (*wc-node*), *light splitting node* (*split-node*), and *virtual source* (*VS*). A DaC-node is capable of tapping a small amount of optical power from the wavelength channel and transmitting the remainder.

The virtual source is capable of both light splitting and wavelength conversion. The wc-node and split-node are self-explanatory. The VS plays a key role in the construction of a multicast forest. It serves as the branching point. The protocol described in [61] works in this manner.

The protocol described in [62] is much more complex than that in [61]. A VS-rooted multicast approach is proposed. The whole procedure is separated into two phases, namely, the network partitioning phase and the tree generation phase. The key idea is that the network is partitioned into regions based on the vicinity of the VS nodes. Then, every node in the network needs to find the nearest VS from it and establish a connection to the VS. Here, the network partitioning phase ends and the tree generation phase begins. This phase makes use of the connectivity provided in the previous phase, so it becomes faster. The source chooses a VS node – the *primary virtual source* (PVS) – to establish a connection with. All other VS nodes that have destinations connected to them are called *secondary virtual source* (SVS). The PVS and SVSes establish connections to the destinations. The pros and cons are obvious. The main advantage is its short setup time. The main disadvantage is that it requires reservation for pre-established paths in the first phase.

2.4.3 Wavelength Assignment

So far, we have not touched on the wavelength assignment problem. It is the last – but not the least – step of WDM multicasting. The multicast forest (tree) built before cannot work without being assigned appropriate wavelengths.

The wavelength assignment problem has been studied extensively [36][41][52][62][88][90]. Generally, the problem is equivalent to the graph-coloring problem. It is NP-complete hard [36], so only heuristics are feasible. The problem concerns many network properties, such as wavelength conversion, and the number of wavelengths available. Obviously, if sufficient wavelengths are available, there will be no wavelength conflict.

Wavelength assignment may associate with *no*, *sparse* or *full* wavelength conversion. In networks without wavelength conversion, the whole WDM network can be viewed as a stack of independent mesh networks. Each mesh network associates with a wavelength, describing the topology of links where the wavelength is still available. Therefore, the wavelength assignment becomes a problem of fitting the multicast tree onto the mesh network. A kind of search algorithm is proposed to deal with the situation where there are a few multicast trees waiting for assignment [36].

An algorithm based on segments is proposed in [90]. WDM multicasting protocols tend to produce a multicast forest. Removing intermediate wavelength conversion incapable nodes and retaining the associated links divide the whole forest into several segments. Assuming that each link has sufficient wavelengths, the first-fit algorithm [41] can be used to assign the wavelengths.

The algorithm proposed in [88] is also useful in multicast wavelength assignment. First, the multicast forest is broken down into segments as described above. Then, a so-called auxiliary graph is drawn in this way: segments are represented as vertices, and two vertices are linked with an edge if the corresponding segments share some common link. Now, the wavelength assignment problem is transformed into a graph-coloring problem. The constraint is that no two adjacent vertices receive the same color.

2.5 Conclusion

We have provided a comprehensive review in this chapter to discuss the current effort to extend the multicast concept to optical networks. We have covered the main optical multicast concept, optical switches with light-splitting capability, multicast over single-hop networks, multicast over mesh networks, and the challenging algorithms developed for multicast routing in the optical domain.

The use of single-hop optical networks for multicast is relatively straightforward. A good scheduling algorithm is needed to ensure its success.

Multicast over WDM mesh networks is dominated by light-splitting and wavelength conversion capabilities. The constraints in light-splitting and wavelength conversion capabilities cause much complication in supporting multicast. These constraints are likely to stay in the near term. Under the circumstances, the success of optical multicast hinges on the ability to find an intelligent algorithm for multicast routing. A substantial research effort is underway in this area.

A basic assumption used in current multicast routing is that a node has either none or ample light-splitting capability, which may not be the case. A generalization can be made here. The problem of multicast routing in a sparse splitting network where each node has finite light-splitting capability can be formed as a degree-constrained Steiner problem [31][79]. How to integrate wavelength conversion is a challenge.

As mentioned, lightpath is the major transport to support IP over an optical network. The MPLS protocol is used as a signaling means to facilitate lightpath management. Recent advancements in MPLS suggest a Generalized MPLS control. The protocol stack is likely to evolve to be IP/GMPLS/WDM. The issue of multicast over a WDM network is likely to evolve in a similar manner, pointing to the possibility of using GMPLS to embrace optical multicast.

Chapter 3 Simulation Methodology

3.1 Simulator Design and Implementation

3.1.1 Why not ns-2?

ns-2 is a popular discrete event network simulator. It is strong in simulating various network protocols, queuing disciplines, transportation protocols, traffic sources, QoS and many others. However, it does not meet our requirements for several reasons.

First, our purpose is to test the performance of routing algorithms. Our main task is to run routing and wavelength assignment algorithms in a centralized manner. In other words, our main purpose is to test the performance of the algorithms, not the protocols. Most of ns-2's functionality is not useful in this situation. The complex node structure, packet delivery, and TCP/IP suite are not necessary at all for our purpose.

Second, compared with ns-2, a simulator is much simpler. The time required to get familiarized with ns-2 (usually one month is the minimum requirement) is enough to develop a simulator.

Third, when we were investigating ns-2, there was no ready-to-use WDM extension. To develop such capability is not an easy task. In contrast, developing a simulator specifically for our purposes has been much easier.

Finally, there are benchmarks to verify our simulator; thus, the correctness of our simulator can be ensured.

3.1.2 System Design

Our simulator is written in C for higher performance. However, to guarantee the quality of the code, the system design follows the object-oriented concept as much as possible. There are more than 20 classes in the system.

Some are basic data structures, including path, tree, matrix, sparse matrix, set, hash table, queue, random number generator, etc. These data structures are mostly designed for general data types. Thus, callbacks are required in a lot of situations. Some are supportive classes, including Dijkstra's algorithm, Floyd's algorithm, the adjacency matrix, the virtual node representation, etc. Others needed by simulation include the event list generator, the topology generator, the WDM network generator, heuristics, and tools to smooth simulation.

3.1.3 System Implementation

For the classes to meet different purposes, callbacks are widely used for users to tailor the simulator to their special needs. The source code of our entire simulator is available at http://www.comp.nus.edu.sg/~hejun/ding/simulator.rar. The simulator has been developed in Visual Studio .net version. The source code is thoroughly remarked.

The random number generator is worth a mention. The quality of random numbers affects the quality of simulation very much. Therefore, we have decided against using the random number generator in C library. Instead, we have chosen to use the Mersenne Twister, which is available at http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/MT2002/emt19937ar.html.

3.2 Simulation Instance Generation

3.2.1 Topology Generation

The Waxman model

Network topology is randomly created by the Waxman model [11], which is widely adopted for its simplicity and efficiency. The network model is as follows: given an area of $L \times L$, randomly pick N points inside the area; the probability of having a link between any two nodes u and v is $\beta \exp \frac{-d(u,v)}{L\alpha}$, where $d(u,v)$ is the distance between u and v. The generated topology goes through two tests.

Connectivity test

The generated topology may be disconnected, especially when α and β are both small. Hence, in our implementation, we add some code to check the connectivity of the graph; if it is not connected, we add the least number of edges to ensure that it does.

Reduction test

The generated topology goes through the reduction test [30]. In fact, for the complex topology generated by the Waxman model, only simple tests are feasible. In particular, the Terminals of Degree 1 (TD1) test and the Non-Terminal of Degree 1 (NTD1) test are applied. Simply put, we try to make sure that there is no node with a degree of 1. Such nodes can simply be removed from the topology (for non-terminal node) or merged with its adjacent node (for terminal node), without affecting the final routing results.

Traditionally, the reduction test is to simplify the graph and thus reduce computational complexity. However, we choose the alternate way of maintaining the difficulty of the instance: we add an extra edge between the vertex and another randomly chosen vertex.

Cost and delay of edges

The simulator allows users to provide their own functions to decide the cost and delay of each edge. There are two built-in functions for use. First, cost is randomly picked in the set {1, …, 10}, which matches the typical values for costs used in the NSFNET backbone network [81]. Delay is proportional to the distance between two nodes. Second, the cost and delay of each edge is a unit.

3.2.2 WDM Network Generation

Besides topology, several WDM-related properties are added for resemblance to a real WDM network. At the network level, number of wavelengths (numWave), density of LS-capable nodes (lDense) and density of WC-capable nodes (wDense) are required. At the node level, number of splitters (numLS) and capability of each splitter (LScap, i.e., the incoming signal can be split up to LScap number of outgoing signals) are required for an LS-capable node while number of converters (numWC) is required for a WC-capable node. Currently, we only consider the one-to-one full-range converter, which can convert one wavelength to any other wavelength. The parameters and their meaning are summarized in Table 2.

Table 2 Parameters of the WDM network and their meaning

Parameter	**Meaning**
numWave	Number of wavelengths
lDense	Percentage of nodes that have light splitting capability
wDense	Percentage of nodes that have wavelength conversion capability
numLS	Number of splitters that a node may have
numWC	Number of converters that a node may have
LScap	Capability of each splitter

All the splitters are identical. The share-per-node scheme applies to the splitters. That is to say, all splitters of one node are put in a bank and shared by all channels. The same applies to the converters.

3.2.3 Event List Generation

The event list is generated from the classical traffic model, and it fully represents the model. Overall multicast requests arrive abiding the Poisson process with a rate λ. The multicast session lifetime has a mean $1/\mu$. Each node is equally likely to become the source of the multicast request. Given the size of the multicast group, each node other than the source has equal probability to be a destination. We define traffic intensity as $\rho = \lambda/\mu$, as is usually defined in teletraffic theory [73]. To get the appropriate blocking probability, each simulation runs a sufficiently long time. Note that we allow only multicast in our traffic model.

3.2.4 Wavelength Search Scheme

At any moment, each wavelength is utilized differently. Our algorithms take network utilization into consideration. Thus, some scheme is needed to decide from which wavelength the algorithm should start. The choice of the start wavelength affects the final result, just as the choice of the initial terminal does in the cost of the resultant tree. The proposed schemes of wavelength search are:

CONSERVATIVE The available wavelengths are sorted in the order of utilization. The most-utilized wavelength will be tried first.

OPTIMISTIC The available wavelengths are sorted in reverse order of utilization. The least-utilized wavelength will be tried first.

FIXED The wavelengths are searched in a pre-determined order.

RANDOM The wavelengths are searched in a random order.

The available wavelengths will be searched in one order, and the routing process stops when a multicast tree is successfully constructed. There is also an EXHAUSTIVE scheme. It searches every wavelength and finds the optimum result. However, we disregard this scheme as it is too time consuming.

When wavelength conversion is necessary, the same schemes can be used to guide the choice of target wavelength for conversion. In our simulator, the same order is used to pick the start wavelength and the target wavelength.

3.2.5 Optical Component Deployment Scheme

The optical components may be deployed in many different ways:

LS_UNIFORM Each node has equal possibility to be made LS capable.

WC_UNIFORM Each node has equal possibility to be made WC capable.

LS_CRITICAL Only critical nodes may be made LS capable.

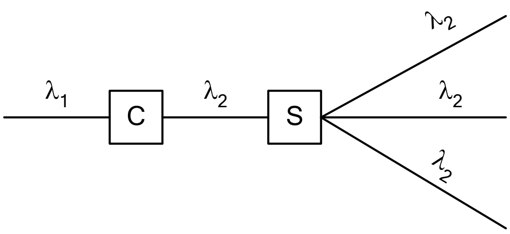
WC_CRITICAL Only critical nodes may be made WC capable.

UNIFORM_VS Each node has equal possibility to be a virtual source.

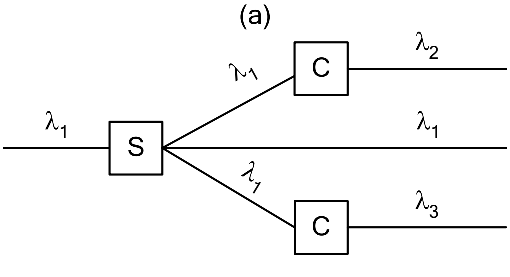
CRITICAL_VS Only critical nodes may be a virtual source.

In our simulator, a critical node is a node that has the largest degree. A virtual source is a node that has both LS and WC capability. Different schemes may be combined. Altogether, there are six valid combinations, namely, CRITICAL_VS, UNIFORM_VS, WC_CRITICAL | LS_CRITICAL, WC_CRITICAL | LS_UNIFORM, WC_UNIFORM | LS_CRITICAL, and WC_UNIFORM | LS_UNIFORM (ordered in priority from high to low).

3.2.6 Optical Component Cascading Scheme



(a)



(b)

Figure 8 Cascading of splitters and converters

Two schemes of deploying splitters and converters can be adopted: CASCADING and STRIGENT (non-cascading). In the CASCADING scheme, one splitter and one converter in the same node (if any) can be used in concatenation in either order. On the contrary, only one splitter or one converter can be used in the STRIGENT scheme. The CASCADING scheme requires more complex design. But it may help increase blocking performance.

Actual cascading patterns are decided by the capability of the optical switch. We restrict cascading to at most two levels for simplicity. That is, there are two possible combinations: converter-splitter or splitter-converter (Figure 8). Figure 8b shows only one possible case of splitter-converter combination. There are many others.

3.2.7 Simulation Instance Generation

There are three files: inst.evt, inst.top and inst.cfg under one directory. Each file is introduced in detail below. All instances under one directory share the same physical topology and event list. Instances under different directories may share the same physical topology and event list as well. This provides for the largest comparability in the simulation results.

inst.evt Stores the list of events created from the parameters (λ, μ, round), where round is the number of events in the list.

inst.top Stores the Waxman topology created from the parameters (N, L, α, β).

inst.cfg Contains multiple entries. Each entry comprises these parameters: *tDense*, ξ, *numWave*, *wDense*, *lDense*, *numWC*, *numLS*, *LScap*, *deploy*, *wsearch*, *cascade*, and *numInst*.

Some parameters have already been defined in Table 2; others are defined in Table 3.

Table 3 Parameters of simulation instance and their meaning

Parameter	Meaning
λ	Multicast request arrival rate
μ	Reciprocal of mean multicast session lifetime
round	Number of multicast requests to be simulated
N, L, α, β	Parameters for the Waxman model
tDense	Percentage of multicast group members
ξ	Delay coefficient, decide the delay constraint for each multicast request
deploy	Optical component deployment scheme
wsearch	Wavelength search scheme
cascade	Optical component cascading scheme
numInst	Control for the number of instances generated from the set of parameters

3.3 WDM Network Specific Routing Issues

3.3.1 Multiλ-light-tree

The light-tree concept was proposed in [52]. The light-tree is restricted by the wavelength-continuous constraint. In other words, only light splitters are used to improve performance. In our approach, we extend the concept to include the use of wavelength converters. We name it the multiλ-light-tree to differentiate it from the light-tree. Adopting the multiλ-light-tree concept enables all-optical transmission. However, to set up the multiλ-light-tree, a proper signaling protocol is needed to adjust the components (light splitters and wavelength converters) and reserve the resources (e.g. wavelengths) along the tree. The Generalized Multi-Protocol Label Switching (GMPLS) protocol suite is a good candidate for the task [9][10]. As the multiλ-light-tree is a kind of light-tree, we will simply use light-tree to refer to the multiλ-light-tree in the remaining part of the thesis.

3.3.2 Odd Situations

Restricted by finite light splitting and wavelength conversion capabilities, some odd situations – such as a virtual node having two incoming links in the multicast tree – are possible and unavoidable.

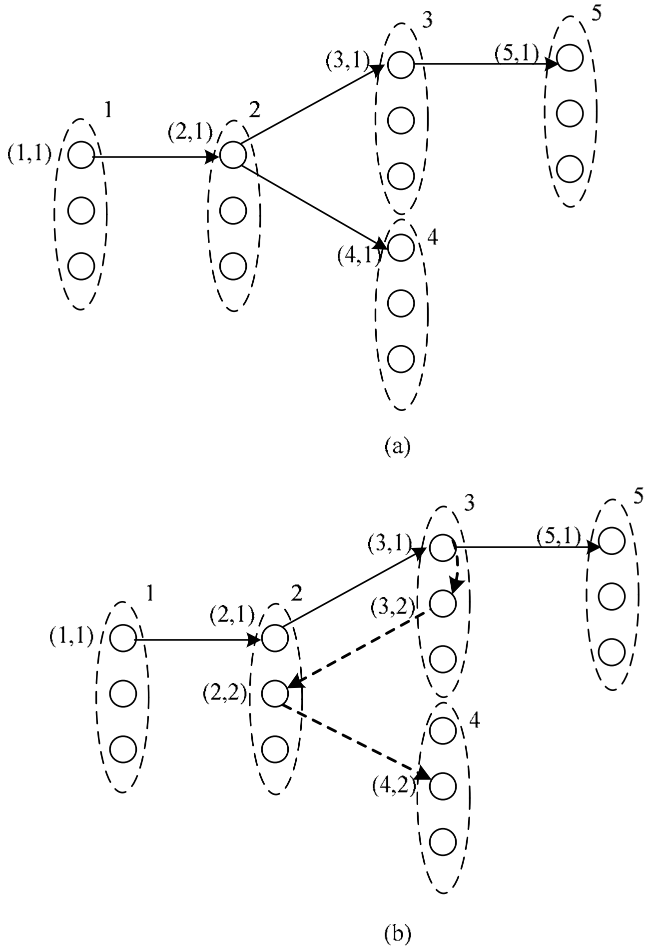


Figure 9 Odd situations in optical multicast

In Figure 9a, node 2 has ample light splitting capability. In such case, one light splitter at node 2 is utilized. However, suppose node 2 does not have any free light splitter. In other words, the node is multicast incapable in the context of optical multicast. In this case, some protocols, such as relative-initiated protocol [3], need to be used to set up a new lightpath to connect the disconnected child node, i.e., node 4. Figure 9b shows a possible scenario. One light splitter and one wavelength converter are utilized at node 3. The incoming signal is first split into two and then one of them is converted to a different wavelength (suppose wavelength 1 at link 3-2 is not available). Node 4 is connected back to the multicast tree through the dashed lines.

At first glance, the resultant multicast tree may not appear to be a tree. However, from the light-tree point of view, the lightpaths connect the source directly to each of the destinations. Therefore, the dashed-line path and the solid-line path actually form two branches of the light-tree. In this sense, the connections form a tree structure.

The above situations are not unusual in wavelength-routed networks. The finite light splitting network causes the unusual situations, and consumes extra resources. In the future, if optical random access memory and quantum computing become available, such odd situations may disappear.

3.4 Benchmarking

We benchmark our simulator against the SteinLib Library, which is popular for the Steiner problem in graphs; it is available at http://elib.zib.de/steinlib/steinlib.php. The test data sets used are I080 and PUC.

The I080 set comprises 100 *incidence* instances. All of them have 80 nodes and various connectivity, from sparse to complete graph. The number of terminals varies from 6 to 20. All instances are fully solved. In other words, optimal solutions are available. We use I080 to benchmark the quality of the solutions.

The PUC set contains three types of instances: hypercube, code covering, and bipartite. Only some of the code covering instances are tested. The other two types of instances are not suitable to our simulator. The chosen instances have the following properties: the number of nodes varies from 64 to 512; the number of edges varies from 192 to 2304; and the number of terminals varies from 8 to 64. The SPH heuristic results are given in [95]. We use PUC to verify the implementation of our simulator.

3.4.1 Benchmarking on the I080 Set

Current state-of-the-art exact algorithms are based on linear programming formulations [95], which take hours or even days to yield optimal results. Thus, they are not feasible for real-time routing. Our simulator runs the SPH heuristic, so its results may be expected to be worse than the optimal. However, the results on all the instances were at most 30% below the optimum. Nine instances even achieved optimal results. Considering the rapid response, the result is quite encouraging.

3.4.2 Test Results on the PUC Set

Table 4 Benchmarking against PUC instances

Name	|V|	|E|	|T|	Opt	SPH_PUC	SPH_SIM
cc6-2p	64	192	12	**3271**	3388	3300
cc6-2u	64	192	12	**32**	32	32
cc3-4p	64	288	8	**2338**	2349	2344
cc3-4u	64	288	8	**23**	23	23
cc3-5p	125	750	13	**3661**	3673	3685
cc3-5u	125	750	13	**36**	36	36
cc5-3p	243	1215	27	*7299*	8266	7790
cc5-3u	243	1215	27	*71*	76	78
cc9-2p	512	2304	64	*17296*	18704	18591
cc9-2u	512	2304	64	*167*	187	185

The test results are listed in Table 4. The *name* column contains the name of the graph. The |V|, |E|, and |T| columns contain the number of vertices, edges and terminals, respectively. The *Opt* column contains the optimum values, but the entries in italic are only the best results so far. The SPH_PUC costs are excerpted from Table III in [95]. The SPH_SIM column contains values attained by our simulator.

Though the SPH heuristic used in our simulator is the same as the one in [95], the costs of the same instances are not identical. However, this is reasonable and to be expected. According to the SPH heuristic, the start vertex is randomly chosen and the ties among the shortest paths of the same cost are broken arbitrarily. We do not know the implementation details of the SPH_PUC simulator. In our case, each graph was tested several times. Each time, the heuristic started from a different terminal and a random number broke the tie, just like tossing a coin. Only the best values are listed in the above table. Our simulator once generated the same result as SPH-PUC for most instances (except the instances cc3-5p and cc5-3u). In conclusion, we have implemented our simulator correctly.

Chapter 4 The Proposed Expanded-Graph Model

4.1 Introduction

There has been active research into the MCRWA problem in recent years. By and large, the studies can be divided into two categories. The first category attempts to solve the problem at the high layer by defining protocols and deploying switches that understand these protocols, and then integrating the high layer and the WDM layer. These protocols either are based on the modification of an existing multicast tree, or use powerful nodes as cores [3]. The second category tries to solve the problem at the WDM layer by utilizing combinatorial optimization algorithms [14]. In particular, this category extensively uses Steiner Minimum Tree (SMT) heuristics, or Linear Programming (LP) methods to construct the multicast tree and perform wavelength assignment [29][47].

Throughout the following chapters, we will use SMT heuristics to solve the MCRWA problem. To fully deploy those heuristics, the WDM network should first be transformed into a suitable graph. Several authors have addressed this problem [21][50]. Here, we seek to add to the body of knowledge on the subject by introducing our new transformation method and providing the simulation results.

We model the MCRWA problem as a tri-criteria problem. The three objectives are: (1) to minimize the use of light splitters, (2) to minimize the use of wavelength converters, and (3) to minimize the cost of the multicast tree. The overall objective is to find a network of minimum cost, subject to the light splitting and wavelength conversion constraints while maximizing the blocking performance of the optical network.

4.2 Definition, Notation and Formulation

4.2.1 Network Topology

Electronic network

Network topology is represented by a simple graph G = (**V**, **E**, c, d), where **V** = $\{v_1, v_2, \ldots, v_N\}$ is the set of vertices, **E** = $\{e_1, e_2, \ldots, e_M\}$ is the set of edges, c: $\mathbf{E} \rightarrow \mathbf{R}^+$ is a real function that assigns each edge a cost, and d: $\mathbf{E} \rightarrow \mathbf{R}^+$ is another real function that assigns each edge a delay. $\boldsymbol{R}^+$ is the positive real set. Each edge can also be denoted by its two ends, e.g. $e(u, v)$. Note that a vertex is also known as a node, while an edge is also called a link.

Functions c and d are both additive. Sometimes, there is a linear relation between functions c and d, that is, $c = \gamma d$. The problem is much simplified by the linear relationship because the shortest path in terms of cost is also the one in terms of delay, and vice versa.

WDM network

Compared with the electronic network, the WDM network has one extra dimension, i.e., wavelength. Thus, a WDM network can be represented by a simple graph **G(V, E,** c, d, **Λ)**, where $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_W\}$ is the set of wavelengths. Only a single-fiber network is considered, and each channel on the fiber supports uni-directional transmission.

Note that both representation forms are in full. There are various simplified forms, such as G = (**V**, **E**, c) when we do not consider delay constraint.

4.2.2 Multicast Group

For a multicast session, let s denote the sender, and $\mathbf{D} = \{d_1, d_2, \ldots, d_F\}$ denote the set of receivers. And, $\mathbf{K} = \{s\} \cup \mathbf{D}$ represents the multicast group, including both sender and receivers. Each element in set **K** is called a terminal or basic vertex.

4.2.3 Network Connection

Paths

Let $P_G(u, v)$ denote a path connecting u and v in graph G. It is simplified to $P(u, v)$ if no ambiguity is caused. Between any pair of nodes, say u and v, there are usually several paths. Among them, three most important paths are the least-cost path (LCP), the least-delay path (LDP), and the delay-constrained least-cost path (DCP). The three paths are denoted as LCP(u, v), LDP(u, v) and DCP(u, v), respectively.

Trees

The basic topology for a multicast communication is a tree, either the Steiner tree, the shortest path tree or the core-based tree. Among all possible trees spanning a certain multicast group **K**, there exist two special trees. One is the minimum-cost tree (MC-tree) whose total cost is minimized, and the other is the delay-constrained tree (DC-tree) where the delay from source to any destination is minimized. The MC-tree and the DC-tree for the same multicast group are usually different. Let MCT_K and DCT_K denote the two trees, respectively.

DC-tree construction can be done in polynomial time by using any shortest distance algorithm with delay as the metric. The most popular algorithm is Dijkstra's algorithm. MC-tree construction is an NP-hard problem, and can be constructed by any appropriate Steiner tree heuristic.

4.2.4 Degree Constraint

The degree-constrained Steiner problem (DCSP) was once used in ATM networks, where some ATM switches could only make a limited number – less than its fan-out degree – of copies of multicast packets [31][79]. By modeling the limitation of the

ATM copying capability into degree constraint, DCSP was applied with some heuristics to solve the problem. Similarly, it can also be applied to address the MCRWA problem. The DCSP problem can be formalized as follows [29][76]:

Given: An undirected network G = (V, E, c), a non-empty set $K \subset V$, and an Integer function $\delta: V \rightarrow I^{+}$, which sets an upper bound on the degree of each node in the multicast tree.

Find: A tree T_K over $K \cup S$, where $S \subset V \setminus K$, such that:

- There is a path between every pair of basic nodes,
- Total cost of edges, $\sum_{e \in T_K} c(e)$ is minimized,
- $\forall i \in K \cup S$, degree of node i, $deg(i) \leq \delta(i)$ in T_K.

I^{+} is the positive integer set. deg(i) stands for the degree of vertex i in graph G. The elements in $V \setminus K$ are called non-terminals. Non-terminals that are included in set **S** are called Steiner vertices.

If the last condition – degree constraint – is relaxed (e.g. let $\delta(i) \geq deg(i)$, $\forall i \in V$), the DCSP problem is reduced to the Steiner Problem (SP). Therefore, it is a generalized form of SP and also an NP-complete problem. The heuristics for the DCSP problem are mostly extensions to those for SP, except that they require the checking of the degree constraint on the resultant multicast tree from time to time.

4.3 Graph Representation of the WDM Network

4.3.1 Multi-Layer Nature of WDM Networks

The WDM network can be viewed as a multi-layer network (Figure 10a). For clarity, we assume that topology is identical between layers. Note that asymmetric topology does not make the problem more difficult. By modeling the WDM network as a collection of wavelength layers, a distinctive dimension – wavelength – is added to the traditional multicast routing problem. The nodes with WC capability serve as ladders connecting different layers (Figure 10c).

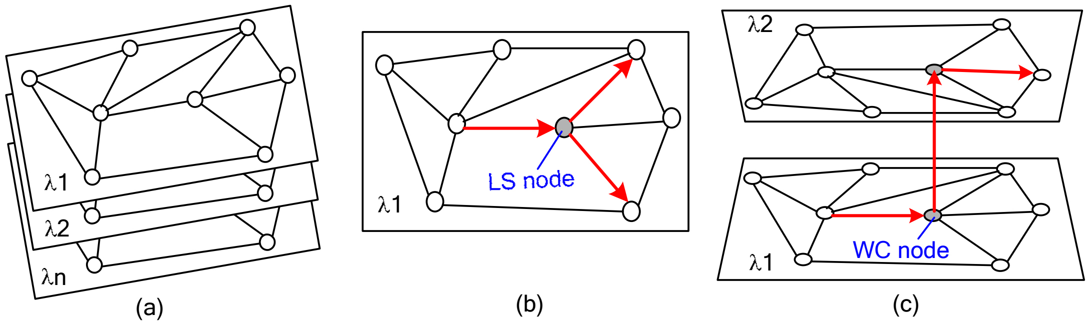


Figure 10 WDM network, light splitting and wavelength conversion

In the WDM environment, the finite LS capability can be transformed to a degree constraint. Generally, if a node has 1:n splitters equipped, then its degree constraint is 1+n. For instance, the LS node in Figure 10b has a 1:2 splitter, so its degree constraint is 3. In other words, the degree of the node in the final multicast tree is at most 3.

Note that degree constraint is not a constant. If a node uses up its splitters, its degree constraint is reduced to 2 (only one incoming and one outgoing connection in the multicast tree). Conversely, if the node has some splitters freed up, its degree constraint resumes.

4.3.2 Layered-Graph Model

The layered-graph model is proposed in [21] for optical unicast routing and wavelength assignment problem. In the model, the optical network consists of core nodes and access nodes residing at the edge. Unicast requests are only initiated by the access nodes. The transformation to graph is straightforward. First, duplicate the optical core topology onto each wavelength. Then split every access node into two parts: source and destination. Finally, connect the two parts to the topology on each wavelength.

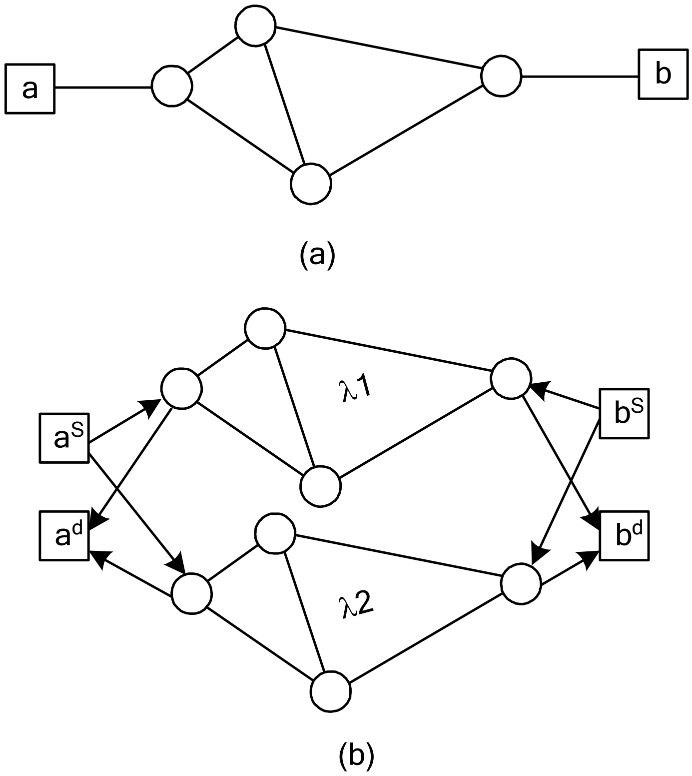


Figure 11 Illustration of the layered-graph model

Figure 11 illustrates the transformation rules. Figure 11a shows the original network topology, where circle-shaped nodes, together with the links among them, form the optical core, and nodes *a* and *b* are the access nodes. Figure 11b is the transformed result (suppose there are two wavelengths). Nodes a^d and a^S are the destination and source parts of the access node *a*, respectively.

The layered-graph model cannot express the wavelength conversion capability of nodes. This is not unusual as the model adheres strictly to wavelength continuity constraint. Furthermore, the access nodes are redundant and can be removed. This is because the access nodes serve two functions: 1) acting as sources and sinks of packets, and 2) ensuring access to all the wavelengths; the first function is trivial while the second function can be substituted by a wavelength-search scheme (cf. section 3.2.4).

4.3.3 The Auxiliary-Graph Model

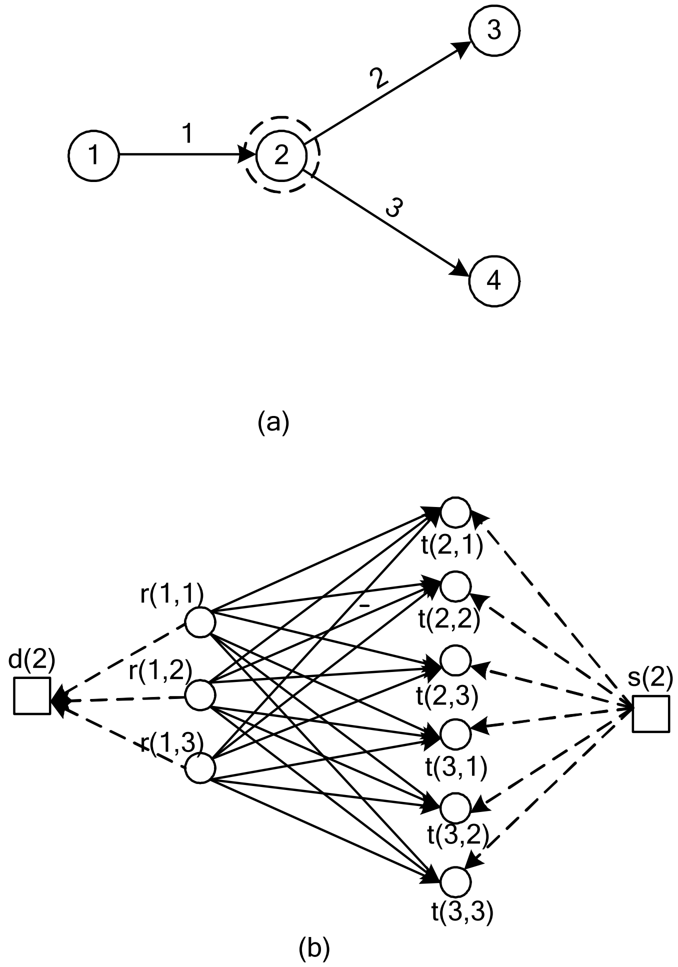


Figure 12 Illustration of the auxiliary-graph model

The auxiliary-graph model is proposed in [50] to deal with unicast in a wavelength convertible network. In this model, each end of a link on each wavelength is represented as a terminal, and each network node is split into two parts – super source and super destination – to act as sources or sinks of packets.

Consider the partial network illustrated in Figure 12a as an example, where each node and link are numbered. Figure 12b shows the transformed result of node 2 (supposing it is wavelength convertible). For example, r(1,3) is the receiving end of link 1 on wavelength 3 while t(2,2) is the transmitting end of link 2 on wavelength 2. d(2) is the super destination node, and s(2) is the super source node of node 2, respectively. A detailed description of the transformation rules may be found in [50].

Even when the other nodes and other directions of traffic are omitted, the resultant graph is quite complex due to the introduction of the receiving and transmitting ends as well as the super source and destination nodes. In the share-per-node scheme [50], all the light splitters and wavelength converters on a node are shared among all connections through that node. Thus, the graph can be much simplified. The super source and destination nodes are not necessary, and all the receiving and transmitting ends on the same wavelength in the same node can be merged into one, as can the converter edges between them.

4.3.4 Expanded-Graph Model

We introduce our expanded-graph model [2] by combining the strengths of both models (i.e., layered-graph model and auxiliary-graph model) and eliminating the weaknesses. Our expanded-graph model mainly simplifies the auxiliary-graph model. It is capable of completing the routing and wavelength assignment in a single step. One fundamental difference is that our model emphasizes on the nodes but not the links.

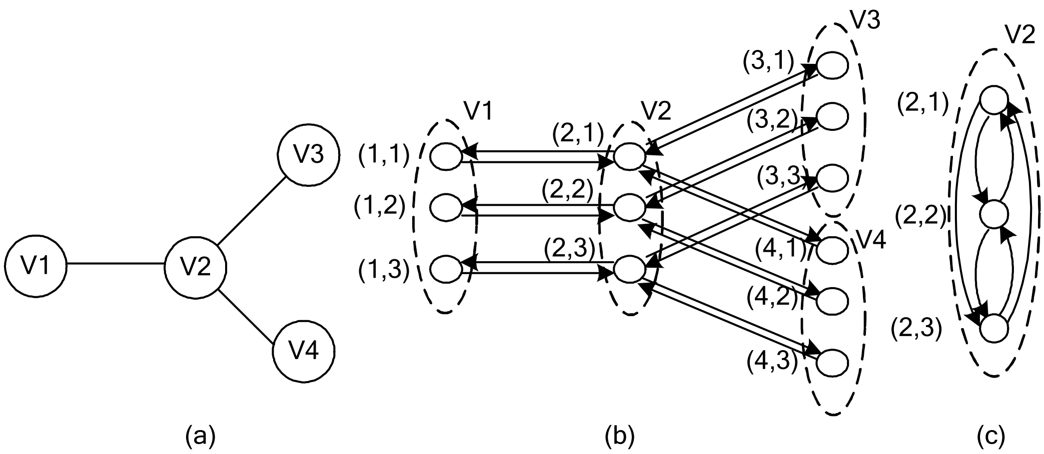


Figure 13 Expanded graph transformation

Each node has a representation on each wavelength, which is called a virtual node and is represented by a tuple (v_i, λ_j) (abbreviated as (i, j)). The topology G is then replicated onto every wavelength. Each edge in G is split into two reverse-directed arcs in the expanded graph. Finally, the wavelength-convertible nodes are processed to connect each wavelength.

Figure 13 illustrates the transformation on the partial network; as shown in Figure 13a, it has three available wavelengths. The transformation follows these steps:

1. Create the virtual nodes

For each node v_i on wavelength λ_j, create a virtual node (i, j).

2. Replicate the topology onto each wavelength

Note that the resultant graph is directed. Now, the graph has changed from Figure 13a to Figure 13b. Here, the layers are disconnected from each other.

3. Deal with wavelength-convertible nodes

The virtual nodes of a wavelength-convertible node have internal links *fully* interconnecting them. For example, wavelength-convertible node V_2 is transformed as depicted in Figure 13c. In addition, if wavelength conversion cost needs to be considered, the cost can be attached to each internal link accordingly.

As described above, the transformation rules and the resultant graph are quite simple. Only the necessary links and nodes are introduced. As shortest path algorithms are sensitive to number of nodes, our model is likely to decrease time complexity as compared with the auxiliary-graph model. In addition, our model can be used in WDM networks, which have limited wavelength conversion capability. Nonetheless, it is suitable only for the share-per-node scheme (refer to Section 3.2.2).

4.4 Routing Heuristic

The expanded graph is a generalized representation of a WDM network. Any SMT heuristics can be applied on the graph after necessary revisions required by the degree constraint. However, in the case of digraph, we prefer the Shortest Path Heuristic (SPH) [29] for its simplicity and efficiency. But to be used in the degree-constrained Steiner problem (DCSP), the SPH heuristic must be modified as required in Section 3.3.2.

Due to degree constraint, all on-tree virtual nodes are divided into two categories: active virtual nodes and inactive virtual nodes. An active virtual node, which abides with its degree constraint, or has free wavelength converters, can be used for tree growth. Conversely, an inactive virtual node would already have reached its degree constraint, and can no longer be used to fork new branch and grow the tree.

```
1. Collect all usable wavelengths into set Γ, and then sort them in some order;
2. while (Γ ≠ φ) {
  2.1 Get the first wavelength λ in Γ, Γ = Γ - {λ}, Σ = {(s, λ)}, T = φ, restore D;
  2.2 while (D ≠ φ) {
    2.2.1 Find the shortest path P(e,d) where d∈D and e∈Σ;
    2.2.2 If (found), add P(e,d) into T, D = D – {d}, adjust Σ;
    2.2.3 If (not found or Σ = φ), goto 2; /* the iteration fails */
  }
  2.3 if (D = φ) algorithm succeeds, stop and output T;
}
3. Algorithm fails;
```

Figure 14 Pseudo-code of the SPH heuristic

All active virtual nodes are stored in a set, Σ. Each time an off-tree destination ($d \in \mathbf{D}$) wants to be added to the tree ($\mathbf{T}$), the shortest path between the destination and one virtual node in the set ($e \in \Sigma$) is calculated using Dijkstra's algorithm [44]. In each cycle, a shortest path is used to add a destination to the tree. This repeats until the algorithm succeeds or fails. In case of the digraph, all paths must start from an active virtual node and ends at an off-tree destination. Hence, the final multicast tree is a directed tree, rooted at the source.

Figure 14 lists the pseudo-code of the modified SPH heuristic. In Step 1, the wavelengths are sorted by one of four schemes described in Section 3.2.4. In Step 2.1, the tuple (s, λ) stands for a virtual node. In Step 2.2.2, the adjustment to Σ comprises two aspects: add active virtual nodes and delete inactive virtual nodes. In particular, if a virtual node abides with its degree constraint or has a free wavelength converter, it is an active virtual node and should be added to Σ. Otherwise, it is an inactive virtual node and must be deleted.

When using wavelength converters, the choice of the converted wavelength depends on Dijkstra's algorithm. In practice, Dijkstra's algorithm is modified according to the wavelength search scheme and the cascading scheme (cf. section 3.2).

The time complexity of Dijkstra's algorithm is $O(W^2N^2)$ because there are WN number of virtual nodes. For the SPH heuristic, in the ith iteration, at most k-i shortest paths have to be computed and compared, where $k = |K|$. Therefore, its worst-case time complexity is $O(k^2W^2N^2)$. The correctness of our heuristic depends on the correctness of Dijkstra's algorithm and the SPH heuristic.

4.5 Simulation

4.5.1 Simulation Methodology

The WDM networks in our simulation all have 100 nodes. Section 3.2 describes the method of simulation instance generation and the parameters. The final multicast tree is uni-directional, going from the source to each destination. The metrics include the average number of hops, splitters and converters in the multicast tree, and blocking performance, which is the most important metric. We have carried out extensive simulations on the SPH heuristic.

4.5.2 Simulation Results

We first test the overall effect of the percentages of light splitting and wavelength conversion. The parameters are set as follows (refer to Table 2 and Table 3 for the meaning of the parameters): numWave = 8, tDense = 30%, numLS = 8, LScap = 4, numWC = 8, λ = 40, μ = 1, N = 100, deploy = WC_UNIFORM | LS_UNIFORM, wsearch = CONSERVATIVE. The wavelength-search scheme is set at CONSERVATIVE because of the superior performance as reported in [1].

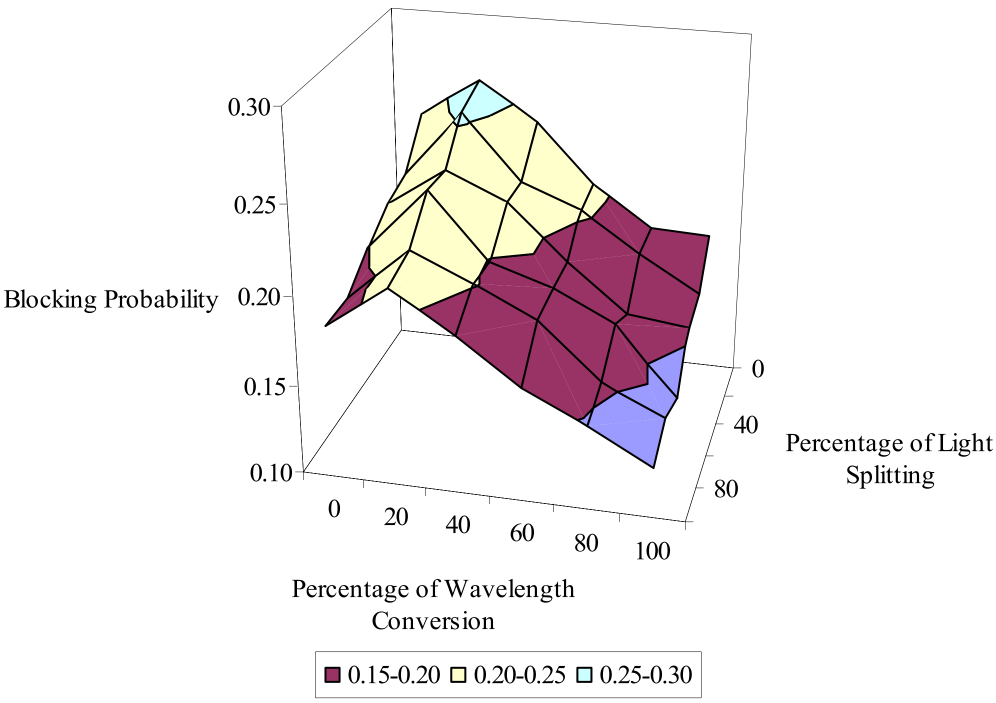


Figure 15 Blocking performance under the STRIGENT scheme

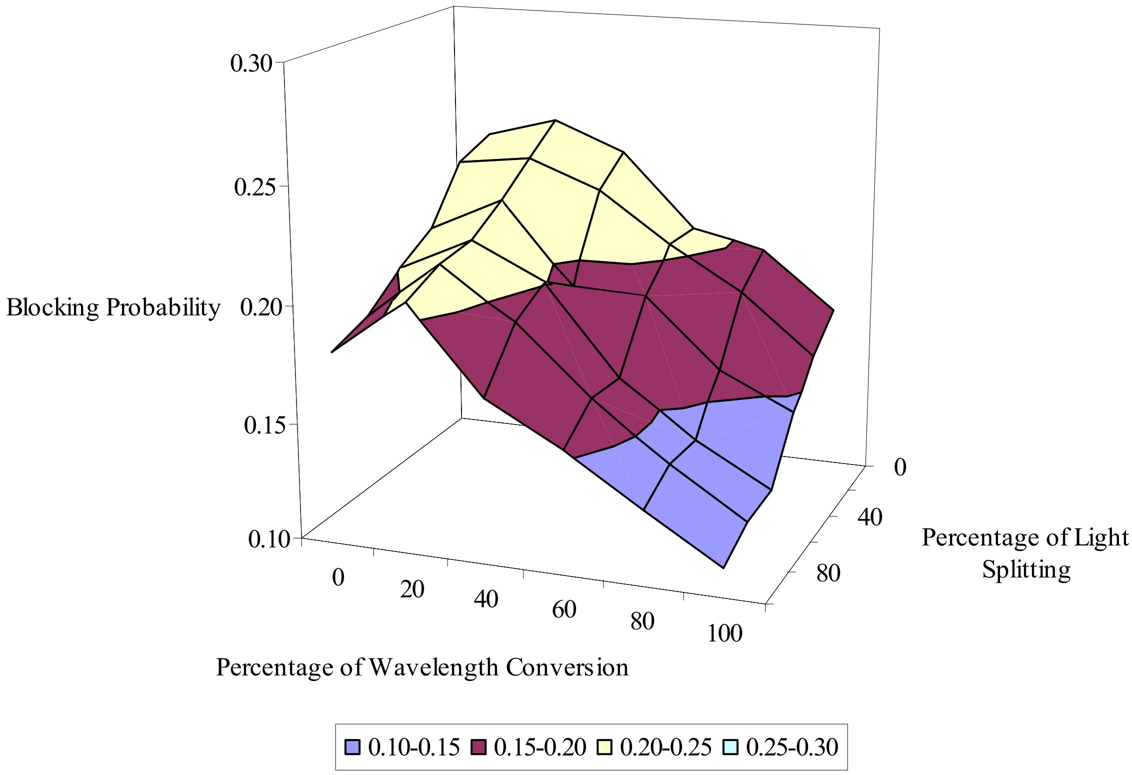


Figure 16 Blocking performance under the CASCADING scheme

As a whole, blocking performance is enhanced with increase in the percentages of light splitting and wavelength conversion under either the STRIGENT scheme or the CASCADING scheme (Figure 15 and Figure 16). The CASCADING scheme slightly outperforms the STRIGENT scheme. This is because the light splitter and wavelength converter can be cascaded under the CASCADING scheme, which brings more chances for successful connections. The increase in wavelength conversion contributes more to blocking performance as compared with light splitting.

Interestingly, the initial increase in wavelength conversion degrades blocking performance, especially in the CASCADING scheme. It seems that the initial increase in wavelength conversion is abused and this in turn interferes with light splitting. That could also be why the CASCADING scheme lags behind the STRIGENT scheme in performance. From this fact, we learn that at least 30% wavelength conversion is required to yield any benefit of wavelength conversion. Otherwise, it is advisable to deploy less wavelength conversion.

Another interesting fact is the half-saddle shape of the average number of hops variation (Figure 17 and Figure 18). The initial increase in wavelength conversion must have been abused, which causes the inclusion of many unnecessary links, and in turn causes the increase in average hops. Referring to the previous two figures, Figure 15 and Figure 16, the initial increase of wavelength conversion degrades blocking performance. Comparing the two pairs of figures, we can clearly see a perfect match. In addition, from the half-saddle shape, we learn that light splitting helps shorten hop distance. This shows the strengths of WDM multicasting again. Finally, it is clear that for shorter hop distances, very dense light splitting with either very sparse or very dense wavelength conversion is required.

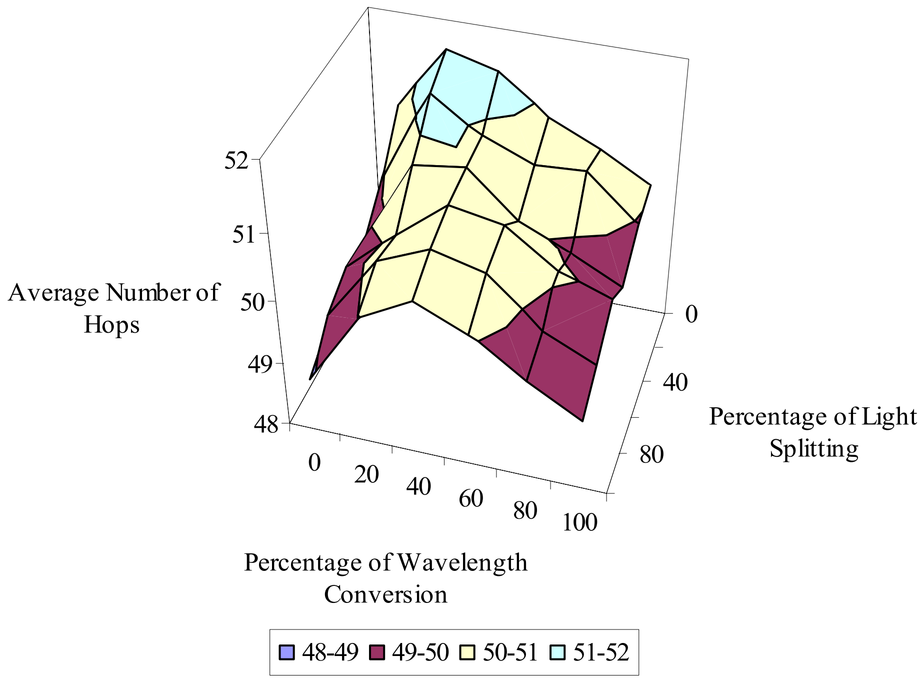


Figure 17 Average number of hops under the STRIGENT scheme

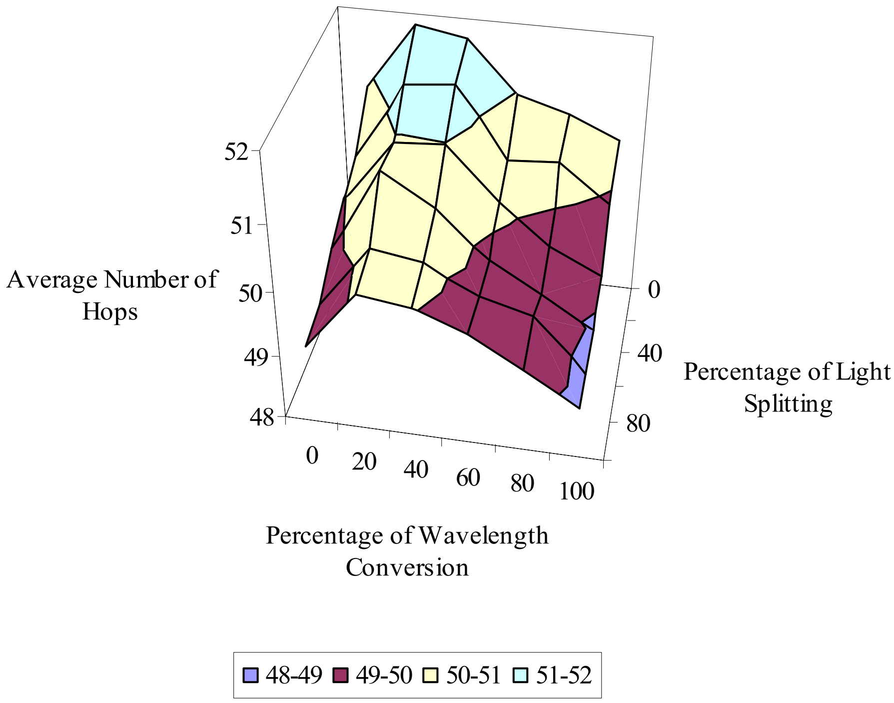


Figure 18 Average number of hops under the CASCADING scheme

Next, we simulate the effect of group size (F) and number of wavelengths (numWave) on blocking performance; the results are presented in Figure 19 and Figure 20. Note that group size and terminal density (tDense) have a simple relationship, $F = N \times tDense - 1$ (cf. section 3.2). The allocation of light splitting and wavelength conversion is fixed and identical in both simulations. The settings for testing the group size effect are:

numWave = 8, lDense = 40%, and wDense = 40%

The settings for testing the number of wavelengths (numWave) effect are:

F = 32, wDense = 40%, and lDense = 40%

Generally, the CASCADING scheme hardly affects blocking performance. With an increase in traffic intensity, blocking performance decreases steadily. Increase in the multicast group size apparently degrades blocking performance (Figure 19). On the other hand, increase in the number of wavelengths can dramatically improve blocking performance. When traffic intensity is low (less than 20), blocking probability remains almost stable in the case of numWave = 8 (Figure 20).

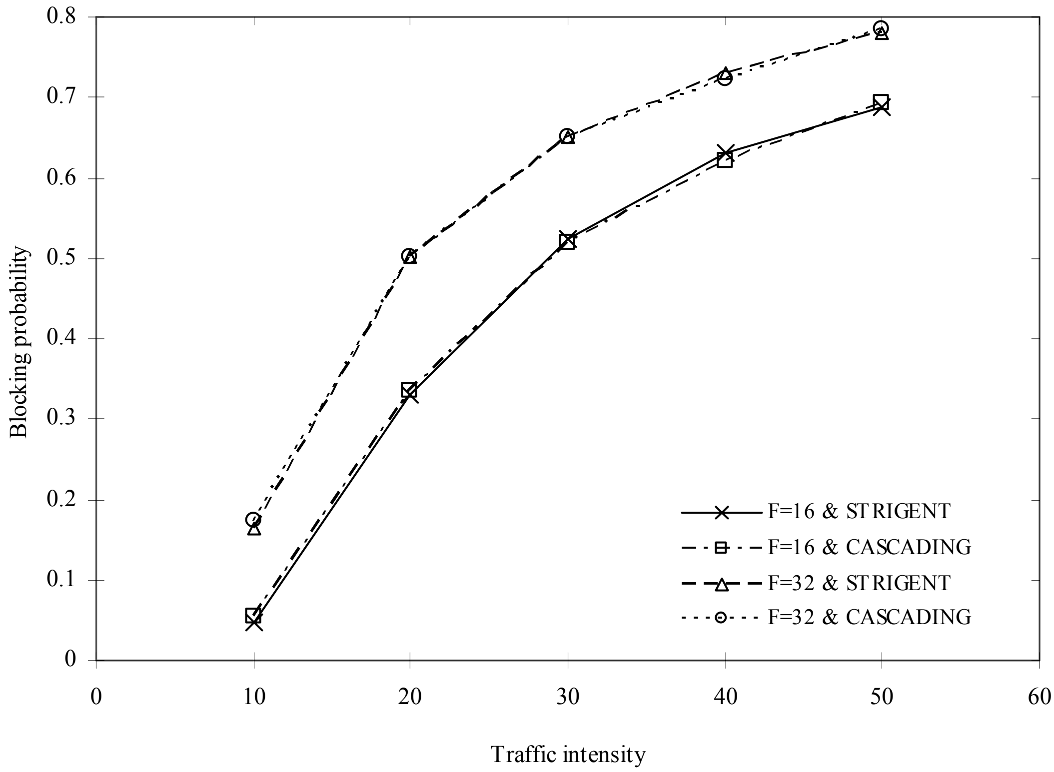


Figure 19 Blocking probability vs. group size

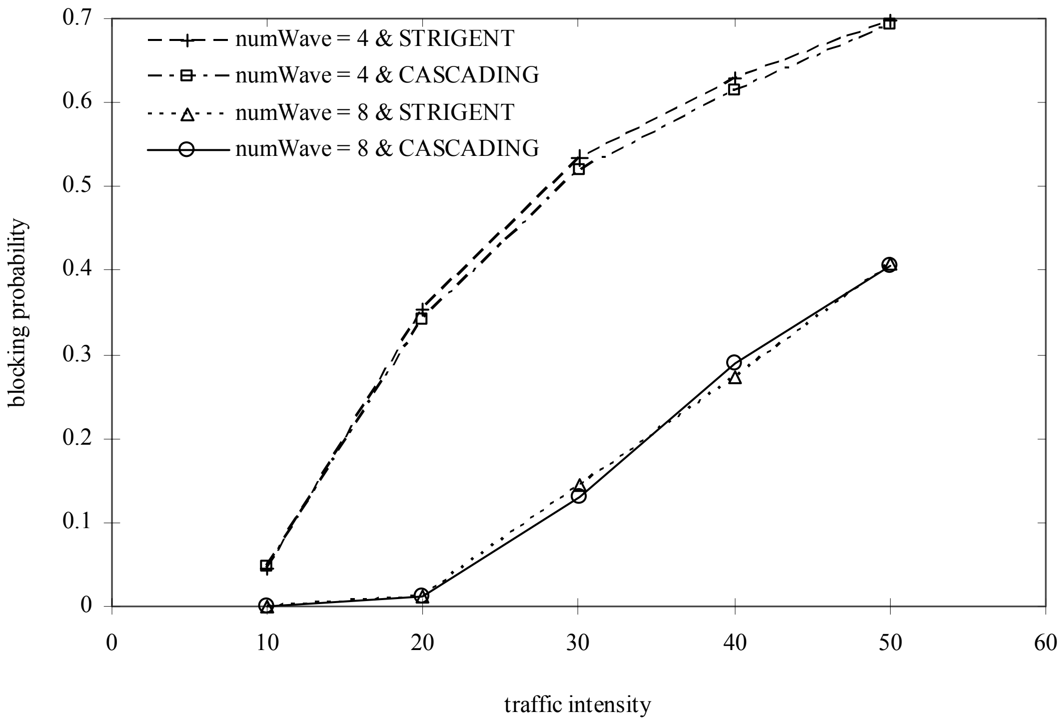


Figure 20 Blocking probability vs. number of wavelengths

To summarize, it is obvious that increase in light splitting, wavelength conversion, and number of wavelengths can improve blocking performance. However, the cost is high, and a tradeoff has to be made between performance and cost. The CASCADING scheme does not show any apparent superiority to the STRIGENT scheme. Considering its higher scheduling complexity, it is not advisable to adopt the CASCADING scheme.

4.6 Conclusions

The expanded graph model has been proposed in this chapter. It is capable of reducing the time complexity of algorithms running on it. However, this model is restricted to the share-per-node scheme, and that can be regarded as its major limitation.

Our simulation has covered only offline multicast, where membership of the multicast is known in advance and fixed in the lifetime of the multicast. It is possible to extend the technique to cover online multicast if desired, as our model is rather general, using Graph Theory algorithms.

In this chapter, we have also applied the degree-constrained SPH heuristic to the model. This approach supports finite light splitting and wavelength conversion capabilities. It is also a rather general approach that could lend itself to wider application.

The simulation results show that a higher percentage of light splitting and wavelength conversion can improve blocking performance. Increase in number of wavelengths can significantly improve blocking performance. These findings are in line with our expectations.

Chapter 5 The Layered-Routing Approach

5.1 Introduction

The protocols and algorithms reviewed in Chapter 2 suffer from a number of drawbacks. First, they handle multicast routing and wavelength assignment separately. Second, although the algorithms do make use of the light-splitting (LS) capability of nodes, they consider only two extreme situations: either full LS capability or none at all. This may not be valid in general. Finally, the algorithms seldom consider wavelength conversion (WC) capability. Even if they do, such as the VS-rooted approach (refer to section 2.4.2), WC is not used in a systematic and efficient way.

The expanded-graph model presented in Chapter 3 finishes routing and wavelength assignment in one single step. It can represent the WDM network correctly. However, it suffers from low computational efficiency. To address the problem, we adopt a layered-routing approach, in accordance to the multi-layer nature of a WDM network (Figure 10). It does routing one layer at a time, and moves to a new layer through wavelength-convertible nodes only when necessary. By doing so, the shortest path algorithms can run much faster, and hence reduce time complexity.

Following the layered-routing approach, we construct two frameworks for developing SMT heuristics [29] in the model. Also, we develop a few heuristics under the frameworks and test their performance through simulations. The SPH (Shortest Path Heuristic) heuristic is modified as a quasi-Prim example, while the ADH (Average Distance Heuristic) heuristic is modified as a quasi-Kruskal example.

5.2 The Layered-Routing Approach

In the WDM model, we represent LS capability as a degree constraint. The additional WC parameter turns the model into a tri-criteria problem that is more complex than the bi-criteria problem discussed in [69]. As a whole, the objective is to minimize blocking probability because the wavelength-routed network is actually a circuit-switched network.

As mentioned earlier, to reduce the time complexity of the expanded-graph model, we introduce the layered-routing approach. Under the approach, routing is done layer by layer, and a new layer is added only when necessary. Here, a layer is synonymous with a wavelength. However, a wavelength may be used more than once in the run of an algorithm. So, we prefer layer to wavelength.

Our approach is able to: 1) finish routing and wavelength assignment in one step, 2) make efficient use of LS and WC capabilities in a more systematic manner, 3) avoid the re-routing procedure all together, 4) model the finite capacity of light-splitting as well as wavelength conversion in a node, and 5) minimize the number of wavelengths used.

We have developed a few heuristics, which can be divided into two groups: quasi-Prim and quasi-Kruskal heuristics. The two groups are similar to the *insertion* and *component connecting* methods with 2Basic building block, respectively [78]. We have also developed other types of heuristics, such as Distance Network Heuristics (DNH), which fall into the category of tree heuristics [29]. The heuristic that we have tested is the simplified DNH by Mehlhorn [51]. However, due to degree constraint, mapping the minimum spanning tree (MST) back into the original graph cannot be done in an efficient manner. We reserve this kind of heuristics for further study.

The quasi-Prim heuristics are similar to Prim's algorithm for constructing MST [45]. They start from a single node (terminal or non-terminal). Every time, a terminal is selected and connected to the tree by a shortest path (not an edge as in Prim's algorithm). The selection criteria for the start point and the terminal to be connected are different from heuristic to heuristic, and comprise the major difference among heuristics.

1. Start the tree with a randomly chosen node (terminal or non-terminal)
2. While there is still a terminal not on the tree
   2.1 If there is no free layer available, the heuristic fails
   2.2 Add a new layer
   2.3 Initialize network topology
   2.4 While there are shortest paths that connect a terminal and an on-tree node
      2.4.1 Choose one shortest path by some rule and add it to the tree
      2.4.2 Adjust the network topology, concerning LS and WC of on-path nodes
   2.5 If the set of off-tree terminals is identical to that when iteration began, the heuristic fails
3. Output the multicast tree

Figure 21 Quasi-Prim heuristic generic flow

Figure 21 lists the generic flow of the quasi-Prim heuristic in pseudo-code form. A new layer will be added in Step 2.2. Once the new layer is added, the remaining steps will operate on that layer only. In Step 2.3, the network topology of the current layer can be initialized according to the usage of nodes and edges in the layer. In Step 2.4.1, different *rules* make up different heuristics. For instance, for the naïve heuristic, the rule is to randomly pick up a path, while the SPH (Shortest Path Heuristic) heuristic will pick the shortest path. In Step 2.4.2, the adjustment is carried out in two parts. First, the degrees of all on-path nodes are checked. For any node that reaches the upper bound, all remaining (not on-tree yet) edges incident to the node will be removed from

1. Start the tree with all isolated terminals.
2. While there is still a terminal not on the tree
    2.1 If there is no wavelength available, the heuristic fails
    2.2 Add a new layer
    2.3 Initialize network topology
    2.4 While there still exists two components that could be connected
        2.4.1 Choose one node with a certain feature (not for K-SPH)
        2.4.2 Connect two nearest components with a shortest path via the chosen node (if any)
        2.4.3 Merge the two components
        2.4.4 Adjust the network topology concerning the LS and WC of on-path nodes
    2.5 Delete dead components
    2.6 If the set of off-tree terminals is identical to that when iteration began, the heuristic fails
3. Output the multicast tree

Figure 22 Quasi-Kruskal heuristic generic flow

the network. Next, one terminal (already on-tree before being connected) of the path is checked to see whether the use of a wavelength converter is needed. If the terminal is connected to the tree but in the previous layer, then a wavelength converter is necessary. In Step 2.5, if no new terminal can be added into the tree during the iteration, it means that there are no nodes available for tree expansion. Therefore, the heuristic fails. Wavelength assignment is straightforward by directly mapping the layer to wavelength.

The quasi-Kruskal heuristics are similar to another MST construction algorithm –

Kruskal's algorithm [45]. They start from a multicast forest which contains all the isolated terminals. At first, every terminal forms a component. Each time, a path (not an edge as in Kruskal's algorithm) which connects two components and meets certain criteria (e.g., not bringing in a loop in the forest, passing through pre-designated nodes, etc.) is picked and added to the forest. The heuristics end when the forest becomes a tree, or all the components are connected as one.

Figure 22 lists the generic flow of the quasi-Kruskal heuristic in pseudo-code form.. In the K-SPH heuristic, Step 2.4.1 is not necessary. In the ADH (Average Distance Heuristic) heuristic, each node is assigned a value according to a pre-defined function *f*: V $\rightarrow R^+$. A node with the lowest *f*-value will be chosen. In Step 2.4.2, in the K-SPH heuristic, two nearest components are connected via a shortest path between them. However, in the ADH heuristic, two nearest components to the chosen node are connected by a shortest path, which passes through that node. In Step 2.5, a *dead* component is one not containing any WC node. It is dead because it can never be connected to the main tree (lack of ladders).

Several algorithms have been developed in either group. In the quasi-Prim group, there are naïve, and SPH and its variants, which are SPH-dV, SPH-dNN, and in particular, SPH-dZ. In the quasi-Kruskal group, there are K-SPH and ADH. The worst-case error ratio of the original SPH and its variants, ADH and K-SPH is tightly bounded to 2 [29]. In other words, the cost of the tree is less than twice of the optimized tree.

5.3 Development of Heuristics

In our modeling, we divide all nodes in a layer into five groups: branch nodes (B-nodes), link nodes (L-nodes), future nodes (F-nodes), dead nodes and free nodes. The first four groups are already on the multicast tree. B-nodes have the remaining LS capability to connect more nodes, and are held in set *B*. L-nodes and F-nodes both have WC capability, and are held in sets *L* and *F*, respectively. The difference between the two is that L-nodes are added in the previous layer, and F-nodes in the current layer. Each node in *L* and *F* has a queue *Q* to hold usable converters. $Q_L(u)$ stands for the converter queue of node *u* in *L* while $Q_F(v)$ stands for the converter queue of node *v* in *F*. The converters in F-nodes can only be used in future layers while those in L-nodes can be used to link branches in previous layers. That is why they are named so. Once the converter queue is empty, the node is removed from the corresponding queue as it can no longer be used to connect different layers. Dead nodes are the remaining on-tree nodes, and they are useless in tree growth. Free nodes are all the off-tree nodes.

By adopting the convention, a tree (or a component in quasi-Kruskal heuristic) is represented by $B \cup L$. Note that $B \cap L$ may not be empty, and B-nodes are to be used first in tree growth, for the sake of saving WC, by fully deploying LS.

In the quasi-Prim group, there are naïve (also known as ARINS in [78], and naïve in [57]), Shortest Path Heuristic (SPH) and its repetitive variants, which are SPH-dV, SPH-dNN and SPH-dZ. Repetitive SPH heuristics generally generate better results than SPH [29][31]. However, the full iteration takes too much time. So we adopt a hill-climbing approach. The cost of a resultant tree is constantly observed. Whenever there is an increase, the algorithm stops. And we believe the *valley* is found. We add a letter *d*, which means *decreasing*, to the name of each heuristic to reflect this feature. In the quasi-Kruskal group, there are K-SPH (also known as KBH in [29], and MST+P in [78]), and Average Distance Heuristic (ADH). We shall give one heuristic in each group as an example.

5.3.1 SPH

Step 1: Begin with a tree T containing any random terminal t. $B = \{t\}$; $L = \varnothing$; if t has usable converter, $F = \{t\}$, and add one converter into $Q_F(t)$.

Step 2: Find the shortest path $P(u, v)$, where u is a terminal and $v \in B \cup L$. If found, add $P(u, v)$ into T, else goto Step 5.

Step 3: If $v \in L$ and $v \notin B$, remove one converter from $Q_L(v)$. If $Q_L(v)$ becomes empty, remove v from L.

Step 4: For every node $w \in P$: (1) check the degree constraint of each node. If $\deg(w) = \mathrm{d}(w)$, delete remaining edges incident to w from G. If $\deg(w) < \mathrm{d}(w)$, add w to B. (2) If w has a usable wavelength converter, add w to F, and add one converter in $Q_F(w)$. Goto Step 2.

Step 5: If there is no terminal left or no wavelength available, goto Step 7.

Step 6: Add a new layer. Restore G. Set $B = \varnothing$, $L = L \cup F$ (add $Q_F(v)$ to $Q_L(v)$ for every $v \in L \cap F$), and $F = \varnothing$ (empty $Q_F(v)$ for every $v \in F$). Goto Step 2.

Step 7: If no terminal left, the algorithm successfully ends, else it fails.

Figure 23 The pseudo-code of the SPH heuristic

The description of the original heuristic can be found as SPH in [29], MPH in [66], and CHINS in [78]. Figure 23 shows the pseudo-code of our modified SPH heuristic. In Step 2, not all nodes in the tree need to be compared. This speeds up the heuristic, especially in sparse LS and WC networks. In Step 3, the condition ($v \in L$ and $v \notin B$) means that v is already on the tree but included in a previous layer. In this case, a wavelength converter has to be used to connect two parts in different layers. Step 4 is the adjustment mentioned in Step 2.4.2 of the quasi-Prim generic flow (Figure 21). In this thesis, we assume the WDM network is *even*, in the sense that every layer is identical in topology. In fact, all heuristics can be used in uneven WDM networks with just a minor change in Step 6: restore the network topology G according to the current layer topology (The same applies to Step 2.3 of the quasi-Prim generic flow in Figure 21).

5.3.2 ADH

The original form of ADH is named HEUM in [78], and ADH in [29], [66]. The function *f*(*v*) for measuring the centrality of node *v* is defined as the proximity of *v* to its neighboring components. Let *C* denote the set of all *r* trees (components) that are sorted in a non-decreasing order according to their distances to *v*, assuming C = {t_1, t_2, …, t_r}. It is obvious that if *r* is fixed to 2, then ADH reduces to K-SPH. The function is defined below:

$$f(v) = \begin{cases} d(v,t_2), v \in t_1 \\ \min_{2 \le k \le r} \left[\dfrac{\sum_{i=1}^{k} d(v,t_i)}{(k-1)} \right], otherwise \end{cases}$$

Each time, a node with minimum *f*-value is chosen to connect two nearest components to it.

Figure 24 lists the pseudo-code of our modified ADH heuristic. In it, some steps are similar to those of SPH. In Step 6, if a component has no L-node, then it is a dead component. The dead component is removed, and all terminals inside are reversed to their original state (i.e., each represents an isolated component) for future expansion. For the next layer iteration, L-nodes from a component would still remain as the same component (even though they might have become disconnected) because their roots are in the same component.

It is possible to use different *f*-functions to generate different heuristics and achieve different effects. For instance, the *f*-function may be:

Step 1: Begin with a forest T containing all isolated terminals. Set $L = \emptyset$. For each terminal t, add t to B, and if t has any usable converter, add t to F and add one converter into $Q_F(t)$.

Step 2: Calculate f-value for each node. Choose a node v with minimum f-value. If found, connect v with the two nearest components (assume u and w are two ends, other than v, of two shortest paths), else goto Step 5.

Step 3: If $u \in L$ and $u \notin B$, remove one converter from $Q_L(u)$. If $Q_L(u)$ becomes empty, remove u from L. Repeat the same operation on v (if v is not a free node) and w.

Step 4: For every node $w \in P$: (1) check the degree constraint of each node. If $\deg(w) = d(w)$, delete the remaining edges (not on-tree) incident to w from G. If $\deg(w) < d(w)$, add w to B. (2) If w has a usable converter, add w to F, and add one converter in $Q_F(w)$. Goto Step 2.

Step 5: If there is no terminal left or no wavelength available, goto Step 7.

Step 6: Add a new layer. Restore G. Set $B = \emptyset$, $L = L \cup F$ (add $Q_F(v)$ to $Q_L(v)$ for every $v \in L \cap F$) and $F = \emptyset$ (empty $Q_F(v)$ for every $v \in F$). For each component C, if it does not contain any node $v \in L$, delete C from T and reverse every terminal $t \in C$ as an isolated component. Goto Step 2.

Step 7: If only one component remains, the algorithm successfully ends, else it fails.

Figure 24 The pseudo-code of the ADH heuristic

$$f(v) = \rho L(v) + \sigma W(v)$$

L(v) equals d(v) – deg(v), and stands for the number of new edges that v can connect. W(v) equals the number of usable converters of node v. ρ and σ reflect the importance of LS and WC. Both parameters should be finely tuned to get the best performance.

5.4 Simulation

5.4.1 Simulation Setup

The simulation is carried out in two parts. First, all the proposed heuristics are extensively tested on various network topologies, and then the best one is chosen. The simulation instances are generated by the methods introduced in Chapter 3. The possible values of the parameters are listed in Table 5. Each instance is generated from one combination of the values. Altogether, we simulate more than 20,000 network instances. As we will show in the next section, the SPH heuristic beats the others.

Table 5 The value fields of the parameters

N	100
L	160
α	{0.2, 0.3, 0.4, 0.5}
β	{0.1, 0.2, 0.3, 0.4}
numWave	16
tDense	{10%, 20%, 30%}
lDense	{10%, 20%, 30%}
wDense	{10%, 20%, 30%}
numLS	16
LScap	{4, 8, 16}
numWC	16

In the second part, the testing on blocking performance is carried out on the randomly generated WDM networks. The setting of the simulation instances is as follows: N = 100, numWave = 16, LSCap = 4, numLS = 8, numWC = 8, tDense = 30%, lDense = 16%, and wDense = 16%. Note that the average degree of the instances is between 6 and 7. Thus, degree constraint will definitely rule.

5.4.2 Numerical Results

First Part: Comparison of Heuristics

Success Ratio and Computational Complexity

Table 6 gives the number of instances that are unable to find a solution. The average success ratio is over 99.7%, which is much higher than those in [31] and [79]. This may be attributed to the presence of wavelength converters. Those terminals that cannot be connected may find their paths in a new layer brought in by converters. In addition, repetitive SPH heuristics seldom fail due to iteration. All failed instances have an average degree less than 7, and mostly below 5. One possible reason for the failure is the poor connectivity that makes degree constraint redundant, rendering light splitter useless.

Table 6 Failed instances (number in brackets indicates the number of network instances tested)

tDense	Naive	SPH	SPH-dV	SPH-dNN	SPH-dZ	K-SPH	ADH
10% (8640)	4	4	0	0	0	5	6
20% (8640)	6	9	0	0	0	19	19
30% (8640)	1	9	0	0	0	25	26
Sum (25920)	11	22	0	0	0	49	51

Owing to the difference in simulation platform and program technology, it is not feasible to compare run time directly. Instead, we consider approximate computational complexity. Generally, the Quasi-Prim heuristic runs ten times faster than the Quasi-Kruskal heuristic.

The Quasi-Prim heuristic depends on Dijkstra's algorithm [44] to calculate the shortest paths between terminals and on-tree nodes. The algorithm needs $O(v^2)$ time (Fibonacci heap is out of consideration), which is dominant in time consumption. For the naïve heuristic, at most k-1 calls to Dijkstra's algorithm are needed to connect all terminals. So its computational complexity is roughly $O(kv^2)$. For the SPH heuristic, in the ith iteration, at most k-i shortest paths have to be computed and compared. Therefore, its complexity is $O(k^2v^2)$. Similarly, the worst-case complexity for SPH-dV, SPH-dNN and SPH-dZ are $O(k^2v^3)$, $O(k^4v^2)$ and $O(k^3v^2)$, respectively. However, in practical simulation, repetitive SPH usually terminates fast after a few iterations.

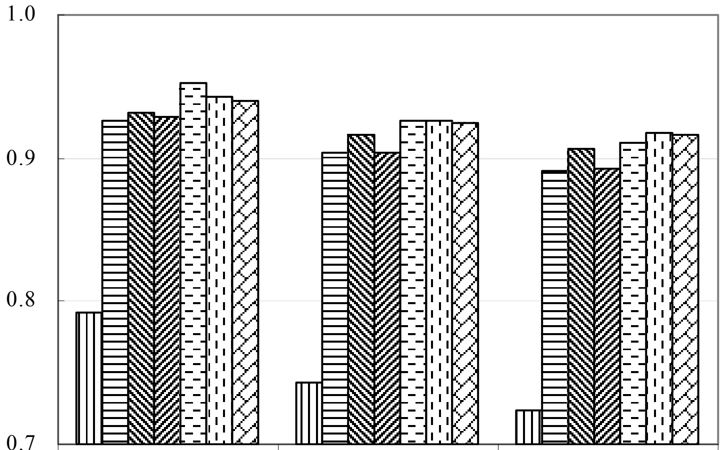
The Quasi-Kruskal heuristic uses Floyd's algorithm [44] to calculate the shortest paths between every pair of components. The algorithm's complexity is $O(v^3)$, and dominant in time consumption. Owing to degree constraint, network topology changes during simulation. In the worst case, the K-SPH heuristic needs to call Floyd's algorithm by k-1 times. Therefore, its worst-case complexity is $O(kv^3)$. Roughly, the worst-case complexity for ADH is also $O(kv^3)$, though the calculation of f-value makes it slower than K-SPH.

Quality of Solution (QoS)

It is difficult to get the absolute optimum result in simulation because network order is prohibitive to exact algorithms. For the appropriate comparison, we adopt the unconstrained naïve heuristic as a reference. Thus, we run the unconstrained naïve heuristic in each network instance without considering degree constraint, and compare all the other heuristics with it. The ratio generally reflects the quality of solution (QoS) of each heuristic. If T_U and T_H denote the trees generated by the unconstrained naïve heuristic and one of our heuristics, respectively, then the QoS is defined as:

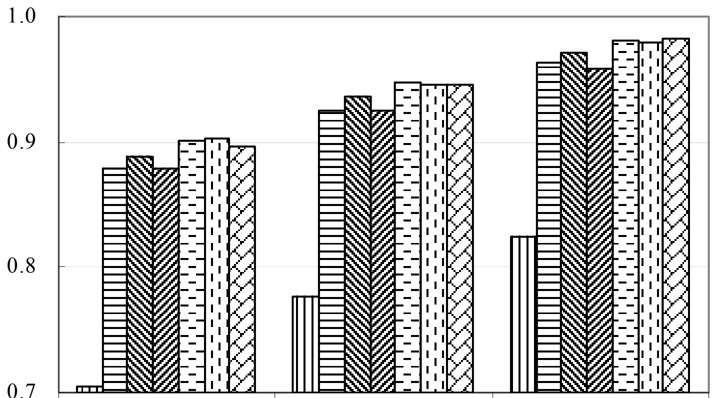
$$\sum_{e\in T_U} c(e) \Big/ \sum_{e\in T_H} c(e)$$

Note that the unconstrained naïve heuristic usually generates the best result, so the ratio is usually less than one. The function c maps edge to cost.



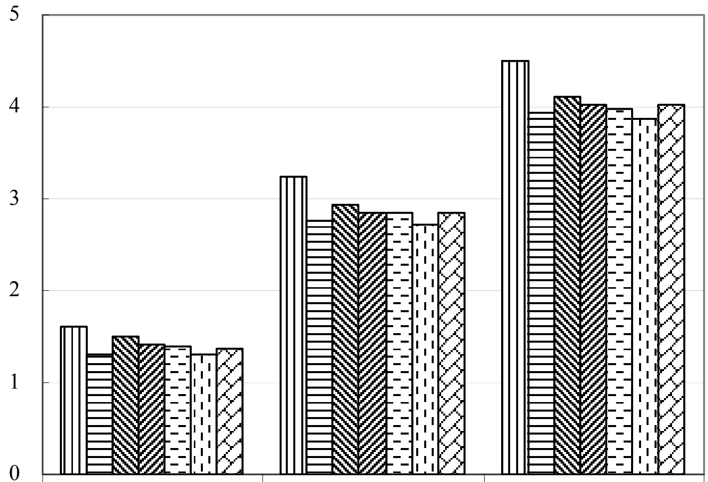
	10%	20%	30%
naive	0.79	0.74	0.72
SPH	0.93	0.90	0.89
SPH-dV	0.93	0.92	0.91
SPH-dNN	0.93	0.90	0.89
SPH-dZ	0.95	0.93	0.91
K-SPH	0.94	0.93	0.92
ADH	0.94	0.92	0.92

Figure 25 QoS vs. terminal density (tDense)



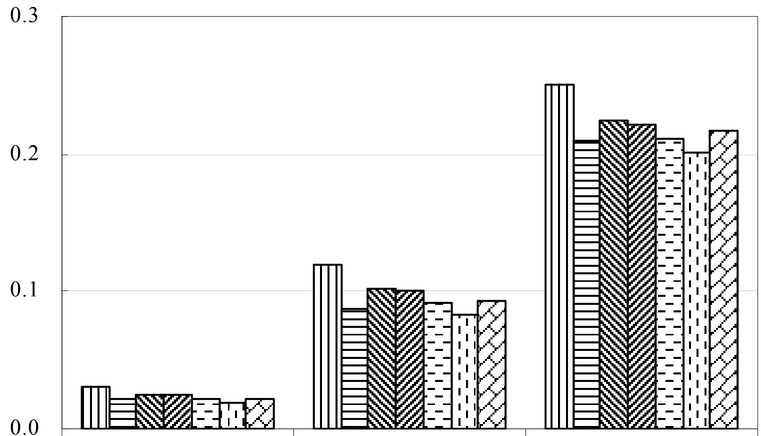
	10%	20%	30%
naive	0.70	0.78	0.82
SPH	0.88	0.93	0.96
SPH-dV	0.89	0.94	0.97
SPH-dNN	0.88	0.93	0.96
SPH-dZ	0.90	0.95	0.98
K-SPH	0.90	0.95	0.98
ADH	0.90	0.95	0.98

Figure 26 QoS vs. LS density (lDense)



	10%	20%	30%
naive	1.61	3.23	4.50
SPH	1.31	2.76	3.94
SPH-dV	1.51	2.94	4.10
SPH-dNN	1.41	2.86	4.03
SPH-dZ	1.40	2.84	3.98
K-SPH	1.30	2.72	3.88
ADH	1.37	2.84	4.02

Figure 27 Number of light splitter vs. terminal density (tDense)



	10%	20%	30%
naive	0.03	0.12	0.25
SPH	0.02	0.09	0.21
SPH-dV	0.03	0.10	0.22
SPH-dNN	0.03	0.10	0.22
SPH-dZ	0.02	0.09	0.21
K-SPH	0.02	0.08	0.20
ADH	0.02	0.09	0.22

Figure 28 Number of wavelength converter vs. terminal density (tDense)

In the system, there are three important factors: tDense, lDense and wDense. An interesting finding is that if we vary one factor and fix the others, the result shows more or less the same trend. Thus, when considering the impact of one factor on one system parameter (e.g., QoS), we will only show the factor and the system parameter. Also, wDense is trivial in a single-session case because only a few instances need wavelength converters (Figure 28).

As a whole, SPH and its variants, K-SPH and ADH produce the most optimized multicast tree. This is expected because the error ratios of the original forms are all bound to 2. The increase of tDense slightly decreases QoS (Figure 25). However, the increase of lDense significantly improves QoS (Figure 26). This is expected because bigger lDense results in more elements in set B, which in turn increases the chance to find a shorter path to connect terminals. The naïve heuristic performs very badly, even though it runs the fastest.

Number of light splitters and wavelength converters needed

We have carried out experiments to investigate the number of light splitters and wavelength converters needed for a network condition. The average numbers of splitters and converters that each heuristic needs are depicted in Figure 27 and Figure 28.

The number of splitters and converters used significantly increases with tDense as more splitters and converters would help absorb the negative effect of the increase in tDense. That is why QoS is stable with tDense (Figure 25). On the average, a multicast session does not need many light splitters and wavelength converters, especially when terminal density is sparse (Figure 27). Except naïve, all other heuristics vary only marginally from each other in the usage of splitters and converters.

In a single multicast session, the need for wavelength converters is trivial.

From the above discussions, one may conclude that except naïve, all other heuristics perform nearly the same under identical conditions. However, if we consider time complexity, Quasi-Kruskal heuristics are obviously out of consideration. In addition, the whole network should be a directional network, indicating that the source must be the origin of the messages. Otherwise, the resultant tree is meaningless in practice. Nonetheless, there is no directional assumption in Graph theory. So, in this case, repetitive SPH heuristics are ruled out. Finally, the suitable candidate is the SPH heuristic, which best balances efficiency of solution and computational effort.

Second Part: Blocking Performance

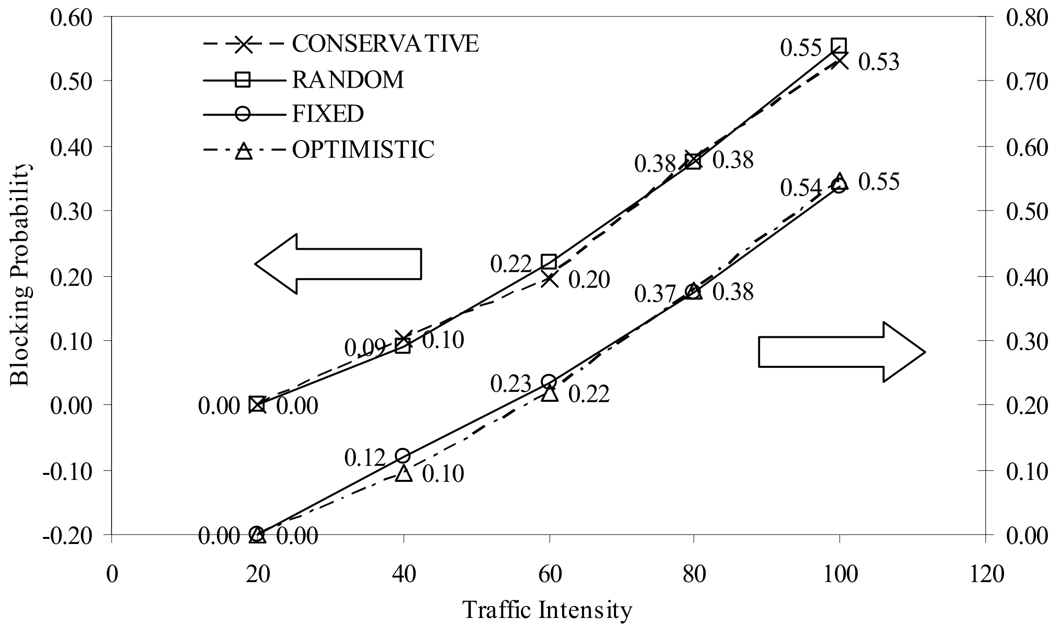


Figure 29 Blocking probability vs. traffic intensity using different wavelength search methods

Figure 29 shows the blocking performance using different wavelength search schemes (cf. section 3.2.4). The optical component deployment scheme is fixed to be LS_UNIFORM | WC_UNIFORM. The cascading scheme is fixed to CASCADING. And numInst = 4. In other words, one set of parameters will be tested four times. All instances are shared with the same WDM network topology.

In the figure, the blocking probabilities of CONSERVATIVE and RANDOM are plotted using the left axis while the other two are plotted using the right axis. The data values of the RANDOM and FIXED schemes are denoted above the corresponding line while the other two are denoted below. From the data values attached to each node, we can learn that CONSERVATIVE is slightly better than the others. However, the difference is marginal. The wavelength search scheme is not an important factor.

Next, we consider the impact on blocking performance under different splitter and converter deployment schemes (cf. section 3.2.5). Through simulation experiments, we hope to know the best way to distribute the devices over the WDM network so as to achieve the best blocking performance with the same budget.

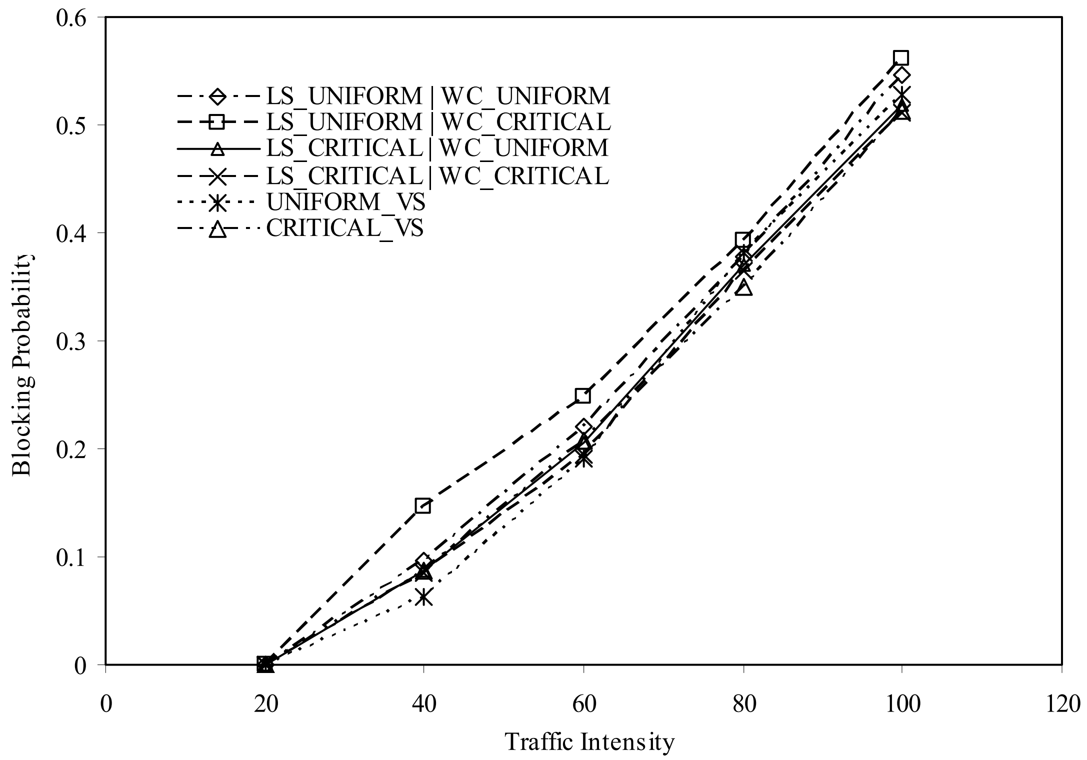


Figure 30 Blocking probability vs. traffic intensity with different splitter and converter deployment scenarios

Table 7 Companion data table for Figure 30

<table>
<tr><th></th><th>20</th><th>40</th><th>60</th><th>80</th><th>100</th></tr>
<tr><td>LS_UNIFORM | WC_UNIFORM</td><td>0</td><td>0.097</td><td>0.220</td><td>0.377</td><td>0.547</td></tr>
<tr><td>LS_UNIFORM | WC_CRITICAL</td><td>0</td><td>0.146</td><td>0.248</td><td>0.392</td><td>0.562</td></tr>
<tr><td>LS_CRITICAL | WC_UNIFORM</td><td>0</td><td>0.086</td><td>0.205</td><td>0.371</td><td>0.519</td></tr>
<tr><td>LS_CRITICAL | WC_CRITICAL</td><td>0</td><td>0.085</td><td>0.195</td><td>0.365</td><td>0.510</td></tr>
<tr><td>UNIFORM_VS</td><td>0</td><td>0.064</td><td>0.190</td><td>0.381</td><td>0.529</td></tr>
<tr><td>CRITICAL_VS</td><td>0</td><td>0.088</td><td>0.207</td><td>0.350</td><td>0.513</td></tr>
<tr><td>Worst Scheme</td><td></td><td colspan="4">LS_UNIFORM | WC_CRITICAL</td></tr>
<tr><td>Best Scheme</td><td></td><td colspan="2">UNIFORM_VS</td><td colspan="2">CRITICAL_VS</td></tr>
</table>

Figure 30 shows the blocking performance for different deployment scenarios under different traffic intensities (refer to Table 7 for data values). The results are obtained using the same conditions as before, plus the CONSERVATIVE wavelength search scheme and the CASCADING scheme. The figure shows that LS_UNIFORM | WC_CRITICAL and LS_UNIFORM | WC_UNIFORM perform the worst in all cases. When the traffic intensity is lower than 60, UNIFORM_VS performs the best. On the other hand, when the traffic intensity is greater than 60, CRITICAL_VS performs the best. This implies that in order to increase blocking performance, the best way is to make nodes virtual sources whenever possible.

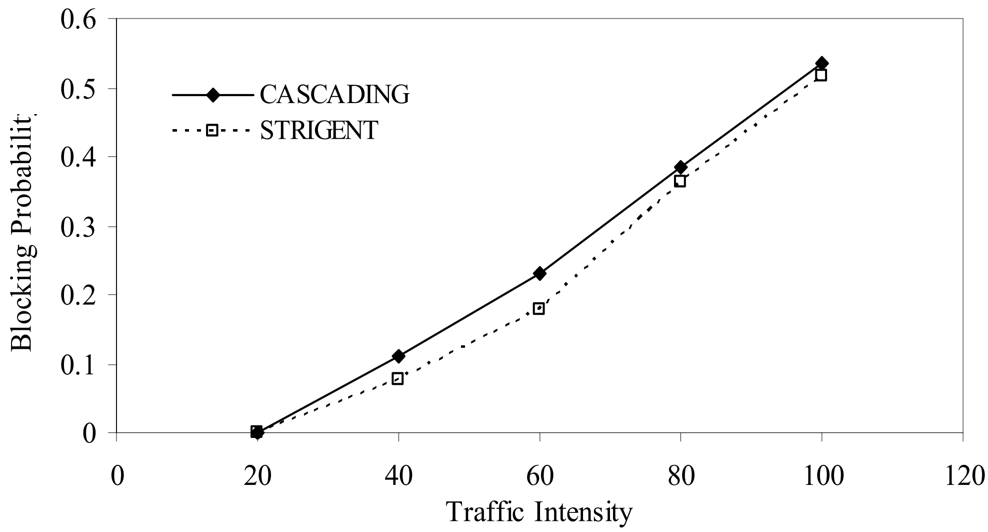


Figure 31 Blocking probability vs. traffic intensity for different cascading schemes

Next, the impact of cascading scheme on blocking probability is simulated. The wavelength search scheme is fixed to be CONSERVATIVE, and the optical component deployment scheme is fixed to be CRITICAL_VS.

Figure 31 shows the simulation results. Unfortunately, contrary to what we have expected (cf. section 3.2.6), the CASCADING scheme performs worse than the STRIGENT scheme. The explanation lies in the finite light splitting and wavelength conversion capabilities. The CASCADING scheme tends to use up light splitters and wavelength converters much faster. Thus, in later simulation runs, some nodes would have used up their capabilities and become normal nodes.

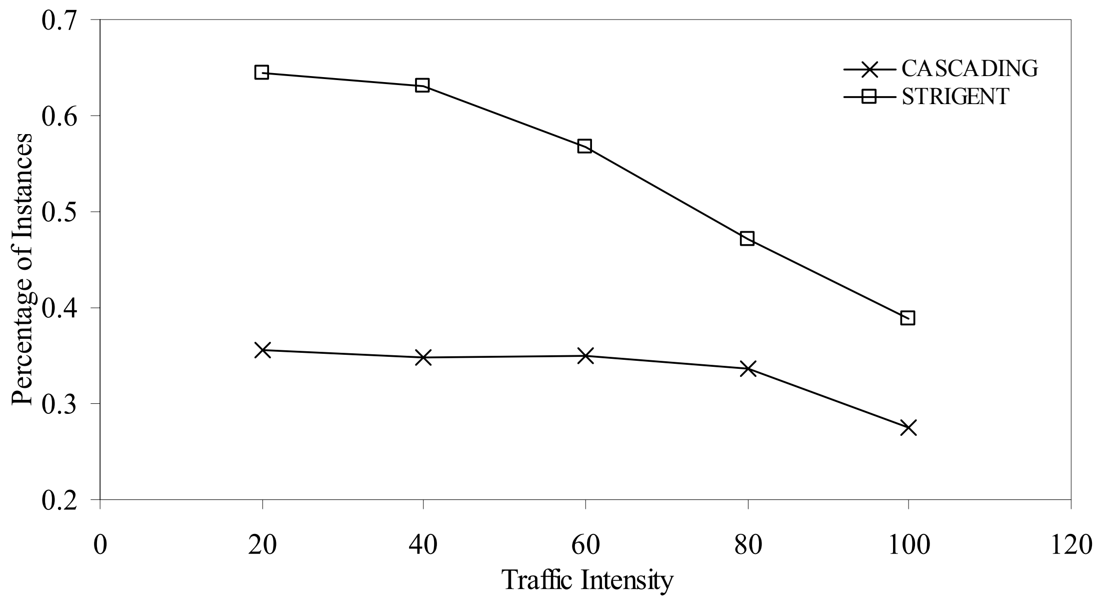


Figure 32 Percentage of instances with minimum cost

Figure 32 shows a different metric. The X-axis represents traffic intensity while the Y-axis represents the percentage of instances that generate the best result under a cascading scheme. Figure 32 supports the conclusion drawn from Figure 31. It is clear that the STRIGENT scheme generates much better results than the CASCADING scheme does. Summarizing from all the results, the STRIGENT scheme is highly recommended. It does not require complex design and implementation, but it generates better results.

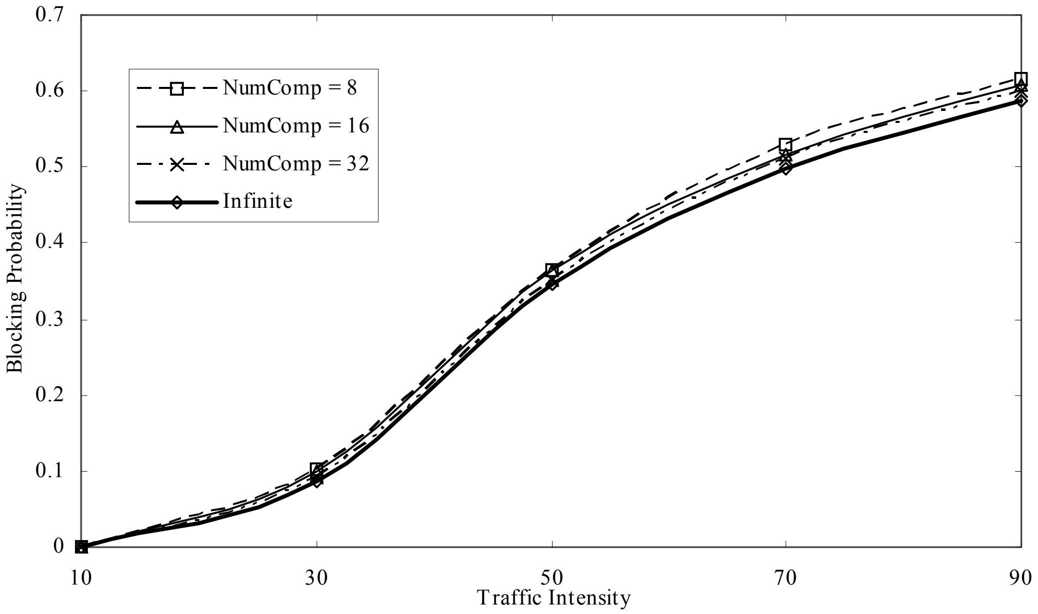


Figure 33 Impact of the number of optical components on blocking probability

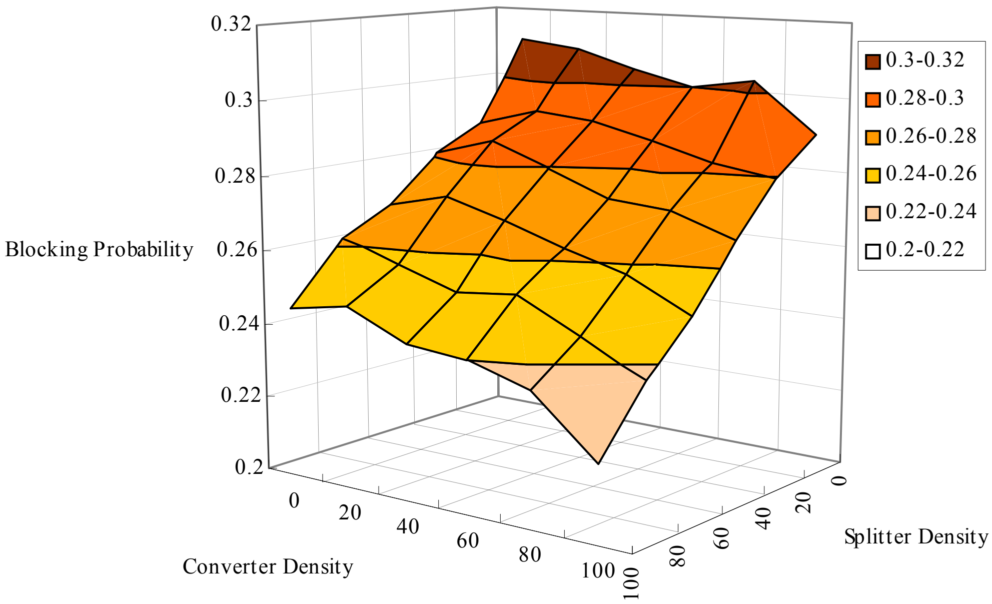


Figure 34 Impact of converter and splitter density on blocking probability

It is obvious that each node does not need to equip an infinite number of optical components to achieve the best blocking performance. Figure 33 shows the impact of the number of optical components on blocking performance. During simulation, lDense = wDense = 100%, tDense = 20%, numWave = 8, and LScap = 8. Each node has equal numbers of splitters and converters. Blocking probability drops when the number of optical components increases. However, the gain from the increase in optical components tends to saturate. Thus, the blocking probability of "NumComp = 32" is very close to the blocking performance of "Infinite" number of components.

Finally, we observe blocking performance when tDense and lDense change (Figure 34). Traffic intensity is fixed at 40, and the other conditions are kept the same. It is clear that when the density of splitters and converters increases, blocking performance improves. However, the contribution from light splitters is somewhat greater than that from wavelength converters.

5.5 Comparison with the Expanded-Graph Model

It seems that the layered-routing approach reduces time complexity and degrades the quality of solution, compared with the expanded-graph model. However, this may not be the case.

The expanded-graph model generates WN number of virtual nodes for an N-node WDM network that has W wavelengths. Dijkstra's algorithm over the graph has $O(W^2N^2)$ time complexity. On the other hand, the computational complexity of the same algorithm in the layered-graph model is only $O(N^2)$. However, time complexity has only limited meaning because it is the worst-case complexity and even the same routing algorithm calls Dijkstra's algorithm different number of times in different models.

On the other hand, the expanded-graph model allows full access to every node and every layer, so it precisely represents the WDM network. For this reason, it seems that the expanded-graph model generates better results than the layered-model. However, this may not always be the case. The WDM network is a circuit-switched network.

Resources along the way are reserved and held during the lifetime of the session. Hence, the way that resources are allocated can heavily affect the success probability for the next request.

The two models are compared by extensive simulations. The same SPH heuristic, with only the necessary minor changes, runs over each model. In the simulation, the blocking performance and run time of both models are compared.

5.5.1 Comparison of Run Time

Dijkstra's algorithm needs $O(N^2)$ time when running over a network with N nodes (Fibonacci heap is out of consideration). We denote the complexity as O(Dijkstra). For the SPH heuristic, in the i^{th} iteration, at most K-i shortest paths have to be computed and compared (K equals the size of the multicast group). So, its worst-case time complexity is $O(K^2 \times O(\text{Dijkstra}))$. Hence, in the worst case, the SPH heuristic that runs over the expanded-graph model is W^2 times slower than the SPH heuristic that runs over the layered-graph model.

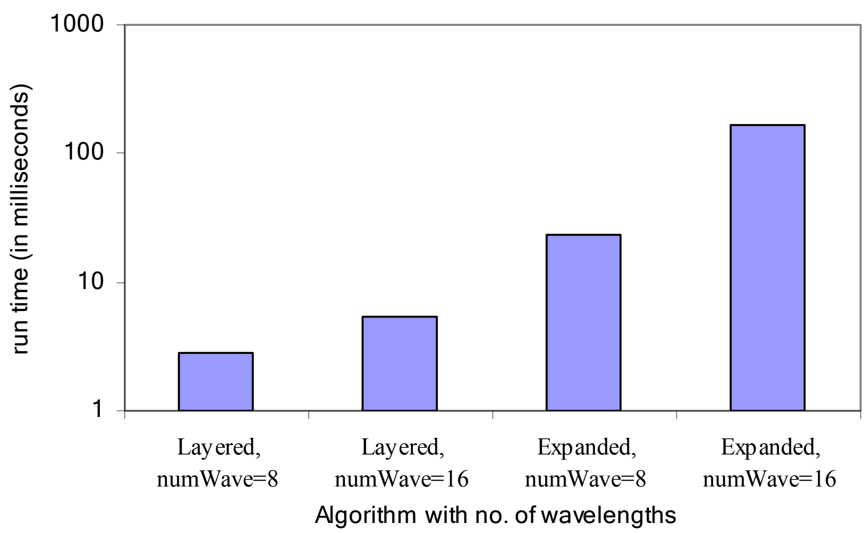


Figure 35 Average run time

Of course, the difference in efficiency of both models is not that high in normal cases. In Figure 35, the label "layered, numWave=8" refers to the WDM network with eight wavelengths transformed by the layered-routing approach. Normally, the layered-routing approach finishes one request in less than 10 milliseconds while the expanded-graph model needs tens of milliseconds. However, on average, the expanded-graph model is much less than W^2 times worse.

5.5.2 Comparison of Blocking Performance

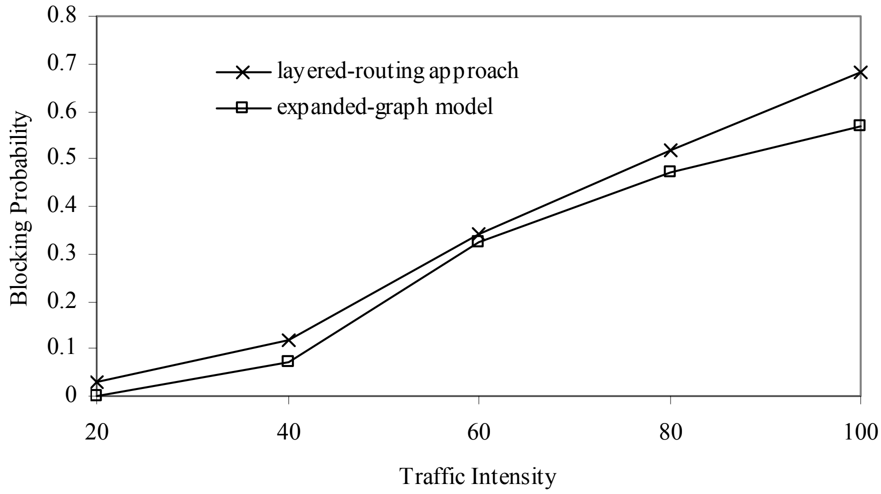


Figure 36 Blocking performance under different traffic intensity

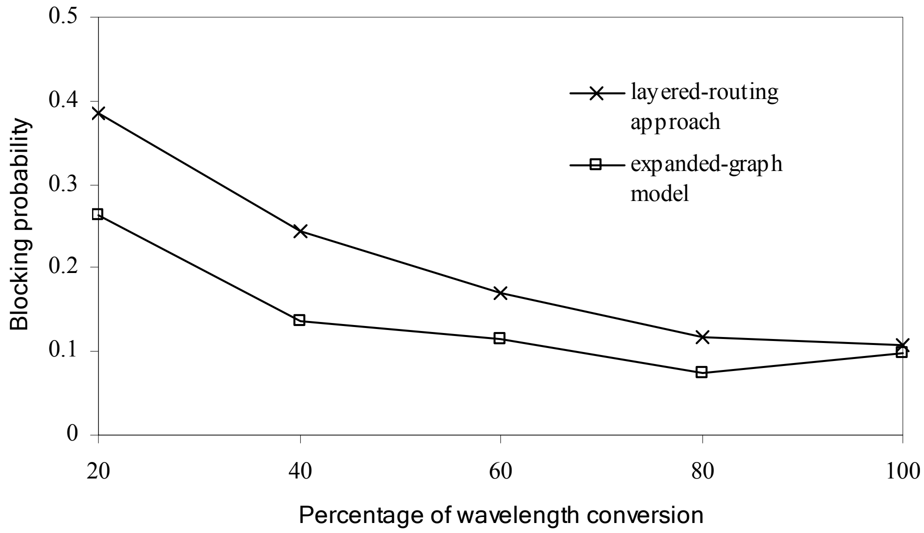


Figure 37 Blocking performance under different wavelength conversion percentages

Figure 36 shows the impact of traffic intensity on the blocking performance of each model. The settings of the simulation instances are numWave = 16, F = 32, and wDense = 80%. Not surprisingly, the blocking performances of both models are not very much apart. The expanded-graph model slightly outperforms the layered-routing approach, especially when traffic intensity is high.

Figure 37 shows the impact of wavelength conversion percentage on the blocking performance of each model. The settings of the simulation instances are numWave = 16, F = 32, and traffic intensity equals 40. With the increase in wavelength conversion percentage, the blocking performance of each model increases steadily. The expanded-graph model outperforms the layered-routing approach again. However, the layered-routing approach gains more from the increase in wavelength conversion percentage.

As expected, the increase in the number of wavelengths and the decrease in the size of the multicast group can improve blocking performance. The figures are omitted to avoid repetition.

5.6 Conclusion

In this chapter, we have developed a new layered-routing approach to address multicast over the WDM network, and applied the DCSP technique to solve the problem. Any 2Basic heuristic can be modified to fit in the approach, and then used to solve the MCRWA problem.

We have tested all the heuristics in a single-session environment. It appears that minimizing the numbers of splitters and converters used will lead to the accommodation of more multicast sessions. The results provide guidance on the use of LS and WC to support optical multicast.

In the simulation, we have not considered the costs of light splitting and wavelength conversion. These two costs can be included in node weight. This is an easy extension, but will increase computational complexity significantly.

A surprising finding is that the STRIGENT scheme performs better than the CASCADING scheme for the layered-routing approach. From Section 4.5.2, we have learnt that the CASCADING scheme only slightly outperforms the STRIGENT scheme. Combining the two findings, the CASCADING scheme can be abandoned to save the time and effort of hardware designers.

We have also compared the layered-routing approach and the expanded-graph model to get a picture of the gain and loss. All in all, the expanded-graph model slightly outperforms the layered-routing approach. However, this is achieved at the cost of much higher time complexity.

Chapter 6 Blocking Performance Analysis

6.1 Introduction

It is costly and time-consuming to construct a WDM test-bed and measure its performance. An analytical model is a very useful alternative. However, blocking performance, especially that of multicast routing, is difficult to analyze theoretically as the corresponding queuing network model is extremely complex and very difficult (if not impossible) to find a solution to.

Researchers around the world have extensively worked on the theoretical analysis of blocking probability, mostly on unicast routing. Girard [8] has compiled a good book on blocking probability analysis in circuit-switched networks. Subsequent works refer extensively to the book. For instance, techniques such as the Brockmeyer model and the Bernoulli-Poisson-Pascal approximation method are used to analyze non-Poisson traffic [76][29].

However, most existing results are on static routing, which includes fixed and alternate routing. In other words, a certain number of routes are pre-calculated and fixed in advance. The routes are searched in a fixed order to find an appropriate one to accommodate the coming request. The overflow model [7] is extensively used to analyze static routing algorithms [1].

Birman [7] proposes a single-link model for the random number of idle wavelengths on a link as a Birth-and-Death process to analyze fixed routing and least loaded routing (LLR). Other authors, including us, have referred to his results frequently. Thus, the model will be discussed in detail below.

The wavelength assignment algorithm heavily affects the analytical model. The available wavelength assignment algorithms include PACK, SPREAD, RANDOM, EXHAUSTIVE and FIXED [1]. The SPREAD algorithm is also known as LLR [7]. The single-link model adopts the RANDOM wavelength assignment algorithm. Hence, it is not suitable for the FIXED wavelength assignment algorithm. Noticeably, edge-disjoint paths are always assumed in the case of alternate routing so as to avoid complex calculations of conditional probabilities. The overflow model, which applies to a route tree, should be used for FIXED wavelength search [1].

Adaptive routing is much more complex. To some extent, adaptive routing uses all possible routes between two nodes as candidates. However, unlike static routing, the routes are not searched in a fixed order. Instead, routing decision is made dynamically on the arrival of a request.

To our knowledge, Hsu et al. [22] are the first to analyze the blocking probability of adaptive routing over WDM networks. They combine the single-link model and the overflow model in their solution. The probability of a route being selected is the key part of the analysis, and the authors consider three different route selection strategies, namely, shortest path, least-loaded path, and weighted shortest path. One drawback of the analysis is that the conditional probability among routes that share common links is not considered.

Overflow traffic (where the peakedness is always greater than 1) may be assumed Poisson (where the peakedness is 1; also known as regular traffic) in most cases. The assumption makes the Erlang loss formula applicable. Though overflow traffic is non-Poisson, the analytical results are normally accurate enough [7]. Moreover, even though the Brockmeyer model or the Bernoulli-Poisson-Pascal approximation can be used to analyze overflow traffic more accurately, the overall results are not necessarily better.

In this chapter, we will first analyze adaptive unicast routing as a warm-up exercise.

Our analytical model extends the one in [22] to analyze WDM networks with limited wavelength conversion. In particular, we will evaluate adaptive routing with the RANDOM wavelength assignment algorithm. We will introduce the analytical procedure immediately in the next section.

Multicast routing has not been well analyzed in WDM networks due to the extreme difficulty involved. To our knowledge, Sahin and Azizolglu [36] are the only to have conducted theoretical analysis on multicast routing. They have analyzed alternate routing, where the candidate multicast trees for each multicast request are decided in advance and searched in a fixed order to choose the first appropriate one when the request actually comes through. The overflow model is adapted to calculate the offered traffic to each link, and then the link decomposition method is used to obtain blocking performance.

In this chapter, we present our analytical model for fixed multicast routing over WDM networks with limited capabilities. Our work represents a major step forward from the existing work; with our model, real WDM networks can now be analyzed.

6.2 Blocking Performance Analysis on Adaptive Unicast Routing

Owing to the fact that not all nodes are wavelength convertible and all wavelength convertible nodes do not have ample capabilities, blocking happens not only on links, but also at nodes. Our model deals with both cases.

6.2.1 Notations, Symbols, and Assumptions

Network and traffic

- The network has an arbitrary topology. Inside the network, there are **N** nodes, **L** links, and **W** wavelengths. Every link has an identical set of wavelengths.

- Each node i (i = 1, 2, … **N**) has $\mathbf{C}_i$ ($\mathbf{C}_i \geq 0$) number of wavelength converters. $\mathbf{C}_i = 0$ indicates a wavelength inconvertible node.
- A lightpath setup request is denoted by two ends (s, d), where s and d are the source node and the destination node, respectively.
- All (s, d) pairs are sorted in topological order. There are at most $N \times (N-1)$ pairs, since all connections are directional.
- The arrival process for each s-d pair i is a Poisson process with rate λ_i. The service time is exponentially distributed with a unit mean.
- The cost of wavelength conversion is ignored for simplicity. We note that wavelength conversion is a costly operation.
- The cost of each link is assumed a unit. Therefore, the cost of a path is equal to its hop distance. This and the above assumption simplify the calculation very much.

Symbols and notations

- For every s-d pair i, the maximum hop distance of all candidate routes is restricted to D_i, and K_i stands for the number of possible routes, the set of which is denoted as R_i under the restriction. In other words, $K_i = |R_i|$.
- P is the abbreviation of the word *probability*. Depending on the context, P can stand for different probabilities. For instance, P_r is the blocking probability of route r. P_B stands for the blocking probability of the whole network.
- The offered load for s-d pair i is T^i (i.e., $T^i = \lambda_i$) while T_r^i is the offered load to route r of s-d pair i.
- One bar (-) is added to the top of the offered load symbol to represent the

corresponding carried load. For instance, $\overline{T_r^i}$ is the carried load on route *r* of *s-d* pair *i*.

- For a link *l*, F_l is the random variable that indicates the number of free wavelengths on that link. The same notation F is also used for segment or path, depending on the context.

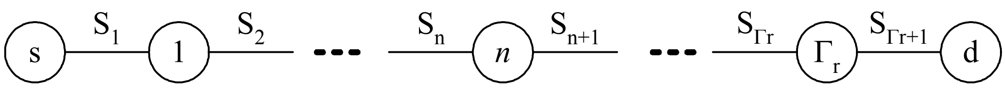


Figure 38 Illustration of segments of route *r*

- The number of interim wavelength convertible nodes of route *r* is denoted by Γ_r (exclusive of nodes *s* and *d*). In turn, the interim wavelength convertible nodes separate the whole route into $\Gamma_r + 1$ number of segments. Figure 38 illustrates. S_i (i = 1, 2, …, $\Gamma_r + 1$) stands for a segment, which comprises wavelength inconvertible nodes only. Therefore, every segment must abide by the wavelength continuity constraint.

Assumptions

- For a single segment, the random wavelength assignment algorithm is adopted.
- Independent link blocking and independent wavelength occupancy are assumed. Due to the fact that not all nodes are wavelength convertible, the two assumptions are not very close to reality. However, for the sake of simple calculation, they are absolutely necessary.
- For every *s-d* pair *i*, only one route *r* with minimum cost will be evaluated. In case the route is blocked, the request will be dropped immediately without the application of overflow procedure.

6.2.2 Analytical Procedure

The candidate routes are divided into segments according to the wavelength convertible nodes on the route. The existing models are adopted to calculate the blocking probability of the routes. The key part is to decide the overflow traffic to the queue of wavelength converters at any wavelength convertible node, and then decide the blocking probability at that node. Finally, the blocking probability of the whole network is calculated from the blocking probabilities of segments and wavelength convertible nodes.

Free wavelength distribution on a single link

The number of free wavelength F_l can be viewed as a birth-and-death process (see Figure 39) [22]. Note that we assume service holding time has unit mean, i.e., $1/\mu = 1$. So the death rate is W-*m* at state *m*.

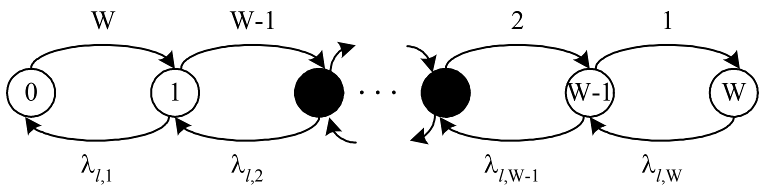


Figure 39 Birth-and-death process for free wavelength distribution on link *l*

By solving the Markov chain, we get:

$$P[F_l = m] = \frac{\prod_{i=1}^{m}(W-i+1)}{\prod_{i=1}^{m}\lambda_{l,i}} P[F_l = 0] \tag{1}$$

$$P[F_l = 0] = \left[1 + \sum_{m=1}^{W} \frac{\prod_{i=1}^{m}(W-i+1)}{\prod_{i=1}^{m}\lambda_{l,i}} \right]^{-1} \tag{2}$$

By definition, the arrival rate $\lambda_{l,f}$ is the summation of all the traffic streams passing through link *l* when it is in state *f*.

$$\lambda_{l,f} = \sum_{i=1}^{N(N-1)} \sum_{\substack{r:l\in r \\ r\in R_i}} T_r^i (1 - P_r[F_l = f]) \tag{3}$$

where $P_r[F_l = f]$ is the conditional probability that route r is blocked when link l has f free wavelengths. Suppose r is an n-hop route, and l is its p^{th} link, then:

$$P_r[F_l = f] = \begin{cases} 1 - \prod_{\substack{i=1 \\ i\neq p}}^{n} (1 - P[F_i = 0]) & if\ f \neq 0 \\ 1 & if\ f = 0 \end{cases} \tag{4}$$

Free wavelength distribution on a segment

The segment must abide with the wavelength continuity constraint. The single-link model [7] is applicable because of the random wavelength assignment algorithm. The result is given directly below (refer to [7] for detailed derivation process). Assuming segment S = {1, 2, …, E}, the probability that segment S has n free wavelengths when each link l of the segment is at its own state F_l is:

$$P[F_S = n] = p_n(F_1,\ldots,F_E) = \sum_{k=n}^{F_{E-1}} p_n(k, F_E) p_k(F_1,\ldots,F_{E-1}) \tag{5}$$

where $p_n(x,y) = \begin{cases} \beta(x,y,n), & if\ x \ge y \ge n, x+y-n \le W, 1 \le x,y \le W \\ \beta(y,x,n), & if\ y \ge x \ge n, x+y-n \le W, 1 \le x,y \le W \\ 0, & otherwise \end{cases}$

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

Therefore, the blocking probability for segment S is simply:

$$P_{seg}(S) = P[F_S = 0] = p_0(F_1,\ldots,F_E) \tag{6}$$

Note that from now on, each segment will be treated as a virtual link. The route will be simplified as a collection of segments as shown in Figure 38.

Overflow traffic to wavelength convertible queue

The converters inside each wavelength convertible node can be viewed as a G/M/C/C queue because the overflowed traffic is not Poisson [8]. For the sake of simplicity, however, we assume the queue is M/M/C/C, and directly apply the Erlang loss formula to the queue. The key is then to decide on the offered load to the converter queue.

The probability that traffic overflows to the converter queue at a node is decided by the two segments adjacent to that node. In particular, the probability is that there is no common idle wavelength across the two segments, on the condition that each of the two segments has free wavelengths (otherwise, blocking happens on the segment). The following formula describes the overflow probability (taking the wavelength convertible node n at route r of s-d pair i as shown in Figure 38 as an example).

$$\begin{aligned} P_{of}^{i,r}(n) &= P[F_{S_n,S_{n+1}} = 0 \mid F_{S_n} > 0, F_{S_{n+1}} > 0] \\ &= \sum_{l=1}^{W}\sum_{m=1}^{W} P[F_{S_n,S_{n+1}} = 0 \mid F_{S_n} = l, F_{S_{n+1}} = m] \\ &= \sum_{l=1}^{W}\sum_{m=1}^{W} p_0(l,m) \end{aligned} \tag{7}$$

where $P_{of}^{i,r}(n)$ stands for the probability that the traffic stream r of *s-d* pair i overflowing to the converter queue at node n. Therefore, the traffic intensity offered to the wavelength convertible node n is:

$$T_{WC}(n) = \sum_{i=1}^{N(N-1)} \sum_{\substack{r:n\in r \\ r\in R_i}} T_r^i P_{of}^{i,r}(n) \tag{8}$$

By applying the Erlang loss formula, the blocking probability of the wavelength converter queue at node n is:

$$P_{WC}(n) = B(T_{WC}(n), C_n) = \frac{(T_{WC}(n))^{C_n}}{C_n!}\left[\sum_{k=0}^{C_n}\frac{(T_{WC}(n))^k}{k!}\right]^{-1} \tag{9}$$

Note that the number of idle wavelength converters can also be viewed as a Birth-and-Death process. However, it is hard to get the state-dependent arrival rate. One feasible solution is to assume the rate to be identical in every state. This method will not be further studied.

Blocking probability analysis of a route

The route is divided into several segments as shown in Figure 38. The blocking of a route happens either on one link or at one node or both. There are many different cases of blocking. It is difficult to calculate blocking probability directly. We will obtain the answer from non-blocking probability instead. Non-blocking probability is the product of the probability that all links are non-blocking and the probability that all nodes are non-blocking. Note that this holds only when the blocking events on all nodes and links are independent.

$$P_r = 1 - \prod_{i=1}^{\Gamma_r+1} (1 - P_{seg}(S_i)) \times \prod_{n=1}^{\Gamma_r} (1 - P_{WC}(n)) \tag{10}$$

Selecting probability of a route

The probability of a route being selected to accommodate a request is decided by the cost formula. In our case, the cost of a path is its hop distance. So the probability of a route being selected is the probability that the route is not blocked times the probability that all the other routes with a shorter hop distance are blocked. This implies that the rest of the routes have longer hop distances than the selected one. The probability of route r from s-d pair i being selected is:

$$\Omega_r^i = (1 - P_r) \prod_{\substack{p:p \in R_i \\ D_p < D_r}} P_p \tag{11}$$

Thus, the offered traffic to route r of s-d pair i is changed to:

$$T_r^i = T^i \Omega_r^i \tag{12}$$

Blocking probability of the whole network

From the route blocking probability, the carried load of each path can be calculated as:

$$\overline{T_r^i} = T_r^i (1 - P_r) \tag{13}$$

$$\overline{T^i} = \sum_r \overline{T_r^i} \tag{14}$$

Finally, the network-wide blocking probability is:

$$P_B = 1 - \frac{\sum_{i=1}^{N(N-1)} \overline{T^i}}{\sum_{i=1}^{N(N-1)} T^i} \tag{15}$$

6.2.3 Iterative Algorithm

As a normal practice, a successive substitution approximation procedure is needed to get the final blocking probability of the whole network. The procedure is briefly described below.

1. Initialization
   - Determine D_i K_i, and R_i for each *s-d* pair *i*.
   - Arbitrarily set initial T^i
   - Assign T_r^i for all *r* and *i*. Pay attention not to violate the traffic conservative rule.
   - Set $P_B^0 = 0$
   - Set number of iterations counter $\delta = 1$
   - Choose an error tolerance value ε. ε should be carefully chosen to give precise enough results at a reasonable cost.
2. Determine the blocking probability of a segment using (1) – (6).
3. Determine the blocking probability of a wavelength convertible node using (7) – (9).
4. Determine the blocking probability of a route using (10).
5. Obtain the route selection probability using (11).
6. Calculate the new values of offered traffic to each route using (12).

1. Pre-calculate the candidate routes (each route is divided into segments) between each pair of nodes, and sort them in the order of their costs. Note that the flat topology without wavelengths is used for calculation.

2. When a request (*s*, *d*) comes

2.1. Retrieve the candidate routes set of the *s-d* pair

2.2. for each route in the set

2.2.1. get the idle wavelength set for each segment

2.2.2. if any of the set is empty, then the current route fails; goto Step 2.2

2.2.3. for each intermediate wavelength convertible node

2.2.3.1. if the wavelength convertible node has no idle converter, then

2.2.3.1.1. calculate the intersection of the idle wavelength sets of the two adjacent segments to that node

2.2.3.1.2. change the sets of the two segments to be the intersection

2.2.3.1.3. in case any idle wavelength set becomes empty, then the current route fails; goto Step 2.2

2.2.4. Now, each segment has its effective idle wavelength set calculated. The wavelength for each segment should be picked from its attached set, taking the wavelength of its previous segment into consideration. Lock the wavelengths and wavelength converters (if any) along the route.

2.2.5. Output the route. Algorithm successfully ends.

2.3. No route is suitable. Algorithm fails.

Figure 40 Pseudo-code for adaptive routing

7. Calculate the overall blocking probability P_B^{δ} using (13) – (15).

8. If $\left|P_B^{\delta} - P_B^{\delta-1}\right| < \varepsilon$, the algorithm stops, and the network-wide blocking probability is obtained. Otherwise, loop back to step 2.

6.2.4 An Adaptive Routing Algorithm

We propose an adaptive routing algorithm based on the theoretical analysis. Unlike other algorithms that separate the procedure into two steps (i.e., unicast routing and wavelength assignment), ours completes them in one single step. The pseudo-code is listed in Figure 40.

To restrict the number of candidate routes and thus computational complexity, the maximum hop distance of the routes can be restricted. The code is from the analytical mode and self-explanatory. Whenever a connection expires, the occupied wavelengths and wavelength converters should be released as well.

6.2.5 Numerical Results

Network topology and traffic model

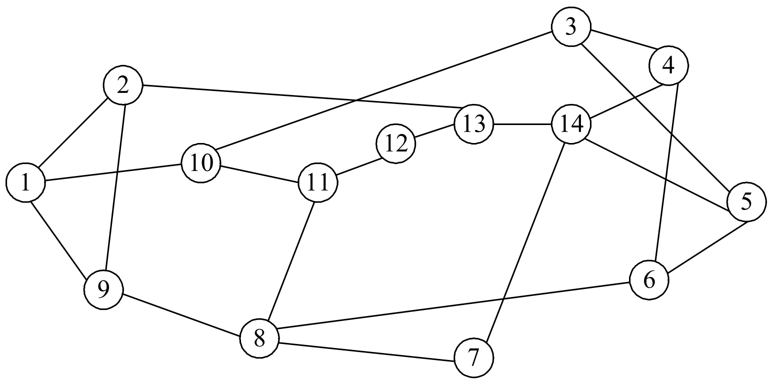


Figure 41 NSFNET topology

The network topology used for simulation is the well-known NSFNET (Figure 41). The main performance metric is blocking performance. The parameters for simulation are set as follows, and the purpose is to save computational power:

- Set $H_i = \hat{H}_i + 1$, where $\hat{H}_i$ is the distance of the shortest path of *s-d* pair *i*. Refer to step 1 in Figure 40.
- The request arrival rate is λ for all *s-d* pairs, and the mean connection holding time is one unit.

- The simulation will run a sufficiently long time to get steady state blocking probability.

Blocking performance

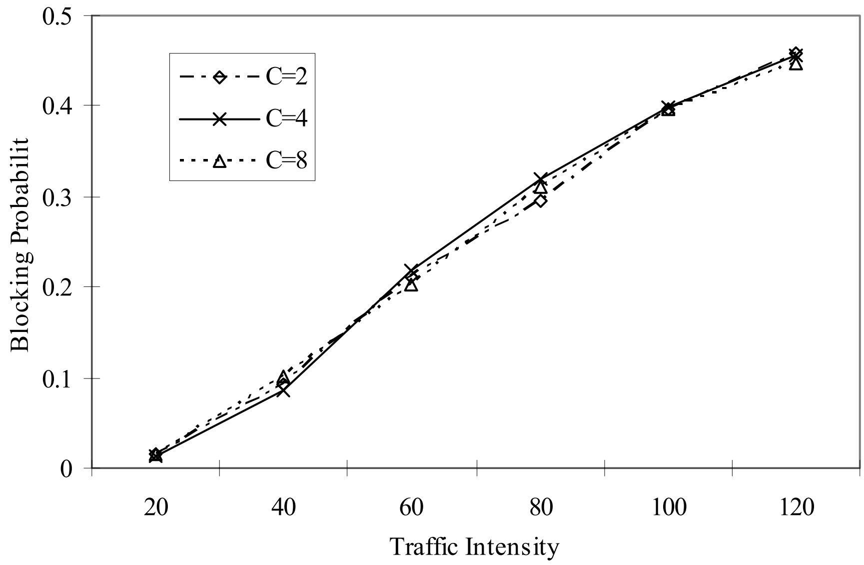


Figure 42 Blocking performance for various number of wavelength converters (wDense = 30% and W = 4)

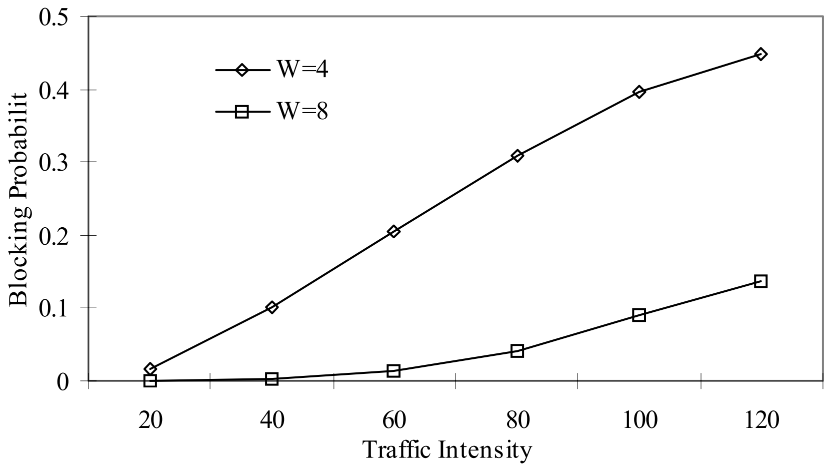


Figure 43 Blocking performance for various number of wavelengths (wDense = 30% and C = 8)

Figure 42 depicts blocking performance variation for various the number of wavelength converters that each wavelength convertible node is equipped with (the number is denoted by C). wDense stands for the percentage of wavelength convertible nodes in the network while W stands for the number of wavelengths of each link.

It is obvious that the number of wavelength converters (C) has very limited impact on blocking performance. Increasing the number of wavelength converters may not necessarily improve blocking performance in some cases.

Figure 43 depicts blocking performance variation for various numbers of wavelengths. The number of wavelengths can heavily affect blocking probability, especially where the traffic load is heavy.

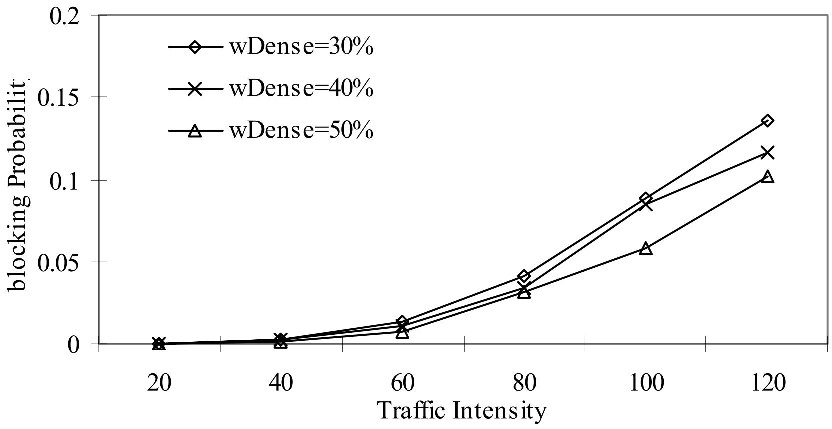


Figure 44 Blocking performance for various density of wavelength conversion (W = C = 8)

Figure 44 shows the impact of wavelength conversion density on blocking performance. Compared with C and W, wDense has moderate impact on blocking performance. The gain on blocking performance is obvious when traffic load is heavy.

In conclusion, among various parameters that affect blocking performance, number of wavelengths (W) is the dominant factor. On the other hand, number of wavelength converters (C) only marginally affects blocking performance. This observation suggests that if the budget is limited, then increasing number of wavelengths is the most effective measure.

The usefulness of wavelength converters depends on the connectivity of the network, and in most cases, only a small fraction of the nodes has to be equipped with wavelength conversion capability for good performance [18]. However, the minimum percentage of wavelength convertible nodes to guarantee good performance is not simulated.

6.3 Analytical Model for Fixed Multicast Routing

The network for analysis can take arbitrary topology. Every node is capable of both wavelength conversion and light splitting. However, every node is restricted to only having limited such capabilities, that is, it is equipped with a limited number of wavelength converters and light splitters. Removing the restriction would only make the problem simpler, as blocking happens only at links.

In general, there are two types of nodes (relay nodes and fork nodes) inside a multicast tree. The type of node is decided by its fan-out degree, F. For instance, node A in Figure 45 is a relay node (F = 1) while node B is a fork node (F > 1). A fork node is much more complex than a relay node, which we have already studied in the previous section.

Each type of node is modeled as a queuing network, and its performance is similarly modeled. Then an iterative method is introduced to calculate overall network blocking probability. As usual, the calculation carries on iteratively, until the blocking probability finally converges.

6.3.1 Notations and Assumptions

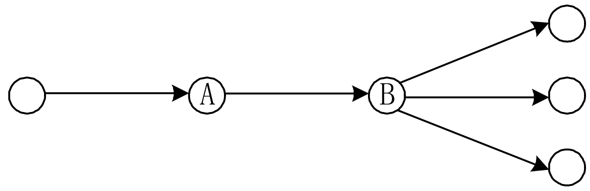


Figure 45 Illustration of fork and relay node

- The network can take any arbitrary topology. Inside the network, there are

N nodes, **E** links, and **W** wavelengths. Every link has an identical set of wavelengths.

- Each node i (i = 1, 2, …, N) has $\boldsymbol{C}_i$ ($\boldsymbol{C}_i > 0$) number of wavelength converters. Similarly, each node i has $\boldsymbol{L}_i$ ($\boldsymbol{L}_i > 0$) number of light splitters.
- A light-tree setup request is denoted by (i, D_i), where i is the source and D_i is the set of destinations. The size of the destination set of a certain node i is fixed to be F_i (i.e., $F_i = |D_i|$). The possible number of destination sets of node i equals the combination $\binom{N-1}{F_i} = \frac{(N-1)!}{F_i!(N-1-F_i)!}$, denoted by S_i. All the destination sets of node i are sorted in canonical order. The j^{th} destination set is denoted as D_i^j. The multicast group is denoted by its source and destination as G_i^j.
- We only consider fixed routing, that is, every multicast group G_i^j has only one fixed multicast tree calculated in advance. The arrival process for the group is a Poisson process with rate ρ_i^j. The service time is exponentially distributed with a unit mean.
- The offered load for group G_i^j is denoted as T_i^j. Due to the characteristic of multicast, the offered traffic to each link is identically T_i^j.
- One bar (-) is added to the top of the offered load symbol to represent the corresponding carried load. For instance, $\overline{T_i^j}$ is the carried load on multicast group G_i^j.
- The cost of wavelength conversion, as well as light splitting, is assumed to be zero, which simplifies the calculation a lot.

- The cost of each link is a unit. That is, the cost of a multicast tree is equal to the number of links inside the tree.
- P is the abbreviation of probability. For instance, P_B stands for the blocking probability of the whole network.
- The set of free wavelengths on a link *e* is denoted as $\Gamma(e)$.
- The wavelengths on a link, and the converters and splitters at a node form three independent queues. The state of a queue is the number of busy servers in that queue. A random variable X is used to denote the state.
- The state of a node is denoted as (m, n), where m and n are the number of busy converters and splitters, respectively. The state of a link is denoted as w, where w is the number of busy wavelengths on the link.
- For a single link, the random wavelength assignment algorithm is adopted.
- Independent link blocking is assumed.
- Independent wavelength occupancy is assumed.

6.3.2 Queuing Network Model for a Node

Distribution of busy wavelengths on a single link

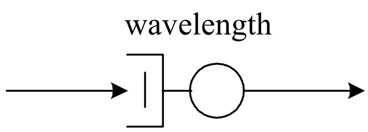


Figure 46 Queuing model for a single link

The wavelengths on a link can be modeled as a queue as depicted in Figure 46. There is **W** number of wavelengths, and there is no waiting space. Therefore, the wavelength queue is denoted as a G/M/W/W queue. Assume the incoming traffic is Poisson with intensity ρ, the probability that there are w busy wavelengths can be decided using the Erlang-B formula.

$$P[X = w] = \frac{\lambda^{W-w}}{(W-w)!}\left(\sum_{k=0}^{W}\frac{\rho^k}{k!}\right)^{-1}$$

The traffic carried by the queue, and thus offered to the downstream adjacent node is:

$$T = \rho \cdot (1 - P[X = W]) \quad (1)$$

Though the incoming traffic to the queue is generally not Poisson (because the peakedness of overflow traffic is greater than 1 [7]), the result is generally accurate, especially when the peakedness is close to 1.

Distribution of common wavelengths among different links

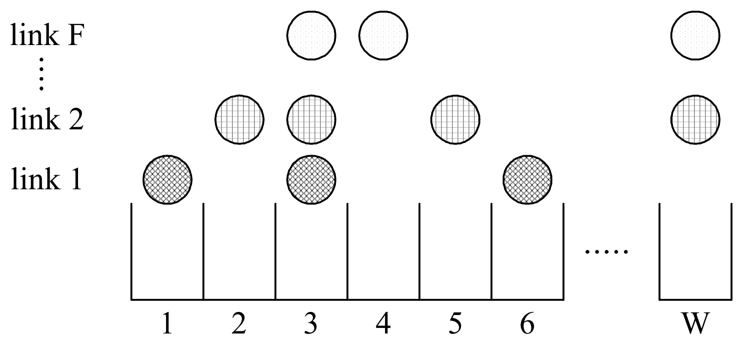


Figure 47 Equivalent model for common wavelength distribution among links

Figure 47 illustrates an equivalent model for the distribution of common (either used or free) wavelengths among different links. Assume there are W cups, each of which is equivalent to a wavelength. Use of a wavelength on a link is equivalent to putting one ball into the corresponding cup. The balls from the same link are denoted using the same color and filling. Then, the probability that altogether *n* cups contain all kinds of balls can be decided as follows:

$$p_n(x_1,\ldots,x_F) = \sum_{k=n}^{x_{F-1}} p_n(k, x_F)\, p_k(x_1,\ldots x_{F-1})$$

where x_i denotes the number of used wavelengths on link *i*, and $p_n(x, y)$ means that there are the same *n* wavelengths occupied when the two links has *x* and *y* used wavelengths, respectively.

The deduction procedure of $p_n(x, y)$ can be found in [7], and the results are listed below:

$$p_n(x,y) = P[X_{i,j} = n \mid X_i = x; X_j = y]$$
$$= \begin{cases} \beta(x,y,n), & if\ x \ge y \ge n, x+y-n \le W, 1 \le x,y \le W \\ \beta(y,x,n), & if\ y \ge x \ge n, x+y-n \le W, 1 \le x,y \le W \\ 0, & otherwise \end{cases}$$

where

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

Of course, to get the result, two implicit assumptions must hold: 1) wavelengths are randomly allocated, and 2) wavelength occupancy is independent among different links.

Queuing network model for a relay node

To a relay node, the light splitting capability is useless. Only its wavelength conversion capability is useful. Figure 48 illustrates the model of a relay node. The wavelength converters inside a wavelength convertible node *i* are modeled as a G/M/C_i/C_i queue. Offered traffic passes through the converter queue with probability α. At the node, the blocking of traffic happens only at the converter queue.

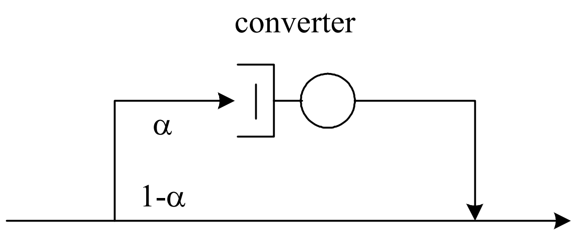


Figure 48 Relay node model

Suppose the upstream and downstream links adjacent to the node are labeled *i* and *i*+1, respectively. Then, we get:

$$\alpha = P[X_{i,i+1} = 0 \mid X_i > 0, X_{i+1} > 0] = \sum_{l=1}^{W}\sum_{m=1}^{W} p_0(l,m). \qquad (2)$$

We refer to the previous sub-section for the deduction of probability $p_0(l,m)$. The implicit assumption here is that direct connection (without wavelength conversion) has higher priority. In other words, the use of a wavelength converter is avoided whenever possible. Therefore, the offered traffic to the converter queue is ρα, where ρ is the offered traffic to the node. The Erlang loss formula is applied to the converter queue.

Queuing network model for a fork node

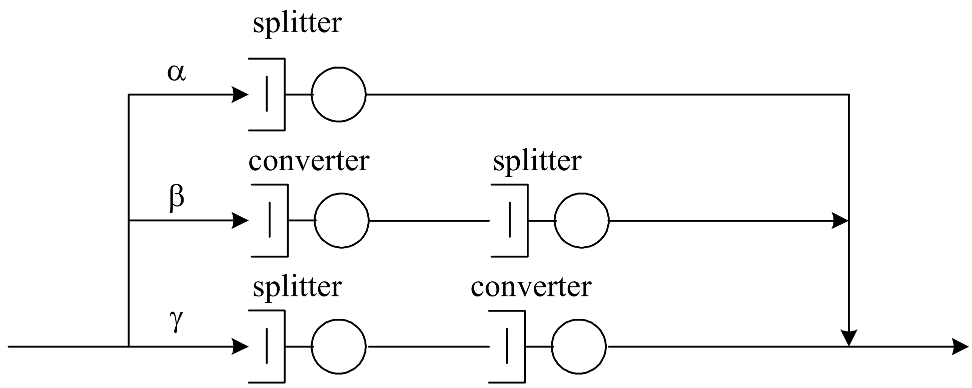


Figure 49 Fork node model

No matter how, the traffic inside a node will definitely go through the queue of either light splitters or wavelength converters. Figure 49 illustrates the queuing network model of a fork node. We allow only two cascading schemes, namely, splitter-then-converter(s) or converter-then-splitter. Altogether, there are three possible scenarios, namely, splitter only, converter-splitter and splitter-converter (ordered by their priorities from high to low).

Note that after the incoming signal is split into several beams of lights, any number of the beams may be required to go through a wavelength converter because of blocking at that outgoing link.

Deciding α, β and γ

Denote the fan-out degree of a fork node as F. The probability of the splitter-only scenario is equivalent to the incoming and outgoing links sharing the same free wavelength. In detail:

$$\begin{aligned}\alpha &= P[\Gamma(0)\cap\Gamma(1)\cap\cdots\cap\Gamma(F)\neq\phi] \\ &= (1-P[\Gamma(0)\cap\Gamma(1)\cap\cdots\cap\Gamma(F)=\phi]) \\ &= 1-p_0(x_0,x_1,\cdots,x_F)\end{aligned} \tag{3}$$

The possibility of using the converter-splitter scenario equals the probability that all outgoing links have at least one common free wavelength and the incoming link does not have any one of the wavelengths. In detail:

$$\begin{aligned}\beta &= P[\Gamma(0)\cap\Gamma(1)\cap\cdots\cap\Gamma(F)=\phi]\cdot P[\Gamma(1)\cap\Gamma(2)\cap\cdots\cap\Gamma(F)\neq\phi] \\ &= p_0(x_0,x_1,\cdots,x_F)\cdot(1-p_0(x_1,x_2,\cdots,x_F))\end{aligned} \tag{4}$$

Obviously, $\alpha+\beta+\gamma=1$. Hence, $\gamma = 1-\alpha-\beta$. (5)

Note that the fork node has a fan-out degree bigger than 1. In the splitter-then-converter scenario (i.e., the incoming signal traverse through splitter first, then the split signals may continue to traverse a converter), some may not traverse the converter queue after exiting the splitter queue. Therefore, the offered traffic to different branches may also be different.

Usage of the queuing models of nodes

The models are very useful for deciding the offered load to the converter and splitter queues inside any node. The key to adopting the models is to decide the offered load to each link and thus the distribution of free wavelengths on each link. Once the multicast tree and the distribution of wavelengths are known, the model, together with the probabilities, can be calculated easily. By summing up all the offered traffic to the converter and splitter queues, the blocking probability at each node can be easily decided by using the Erlang loss formula.

6.3.3 Blocking Performance Analysis of the Whole Network

As usual, the calculation of network-wide blocking probability adopts the iterative manner until the probability finally converges. An iterative algorithm is listed below:

1. Initialization

- Determine the candidate multicast tree for each multicast group G_i^j, where $i \in \{1,2,\cdots,N\}$, and $j \in \{1,2,\cdots,S_i\}$.
- Assign the offered load T_i^j to each multicast tree.
- Set network-wide blocking probability $P_B^0 = 0$.
- Set number of iterations counter $\delta = 1$.
- Choose an error tolerance value ε. ε should be carefully chosen to give a precise enough result at a reasonable cost.

2. For each multicast tree, starting from the root of the tree:
   - Determine the offered load to each link.
   - Determine the type of node and the offered load to the node (i.e., traffic accommodated by the wavelength queue on its previous link) using formula (1).
   - For a relay node, use formula (2) to get the probability and thus the offered load to the converter queue.
   - For a fork node, use formula (3)-(5) to get the probabilities, and thus the offered load to the converter and splitter queues.
3. For any node, sum up the offered traffic, and get the total offered traffic to its converter and splitter queue.
4. Calculate the blocking probability of each queue using the Erlang loss formula.
5. Determine the blocking probability of each node.
6. Till now, the carried load $\overline{T_i^j}$ on each multicast tree can be decided.

7. Calculate overall blocking probability $P_B^\delta = \dfrac{\sum_{i=1}^{N}\sum_{j=1}^{S_i}\overline{T_i^j}}{\sum_{i=1}^{N}\sum_{j=1}^{S_i}T_i^j}$

8. If $\left|P_B^\delta - P_B^{\delta-1}\right| < \varepsilon$, the algorithm stops successfully. Otherwise, loop back to Step 2.

6.3.4 Numerical Results

Simulation methodology

We adopt the ARPANET topology (Figure 50) for simulation. The overall multicast requests arrive following a rate λ Poisson process. Each node is equally likely to become the source of the multicast request, and the multicast session lifetime has a mean 1/μ. Given the size of the multicast group, each node other than the source has equal probability to be a destination. To get the appropriate blocking probability, each simulation runs a sufficiently long time. Note that we consider only multicast in our traffic model.

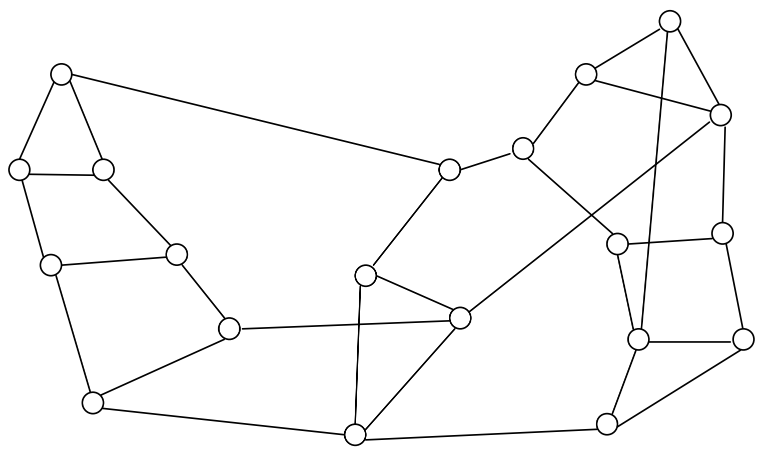


Figure 50 The ARPANET topology

Besides topology, several WDM-related properties are added to form a real WDM network. The properties include the number of wavelengths (W), the number of splitters (nLS) and the number of converters (nWC) a node may have. For the sake of multicast, the size of the multicast group (F, exclusive of the source) is an important parameter.

During simulation, only the three cascading scenarios are allowed. The final multicast tree is uni-directional, only from source to each destination. The metric is blocking performance, which is the most important metric in a circuit-switched network.

Simulation results

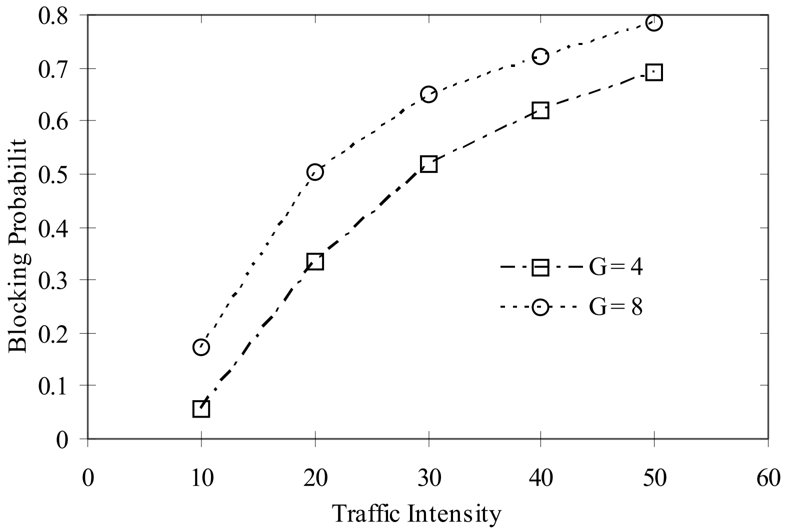


Figure 51 Blocking probability for various group size

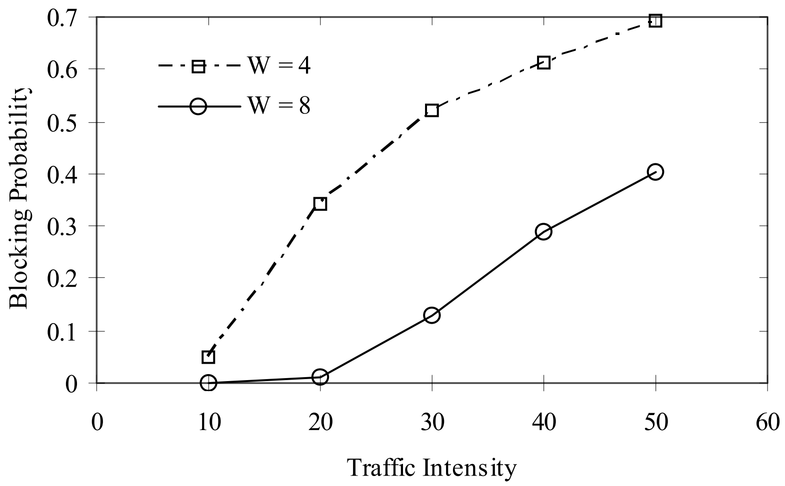


Figure 52 Blocking probability for various number of wavelengths

Figure 51 depicts the relationship between blocking probability and multicast group size. The increase in group size significantly increases blocking probability as well. Figure 52 illustrates the relationship between blocking probability and number of wavelengths. The increase in number of wavelengths can significantly decrease blocking probability. The results here are quite normal and as expected.

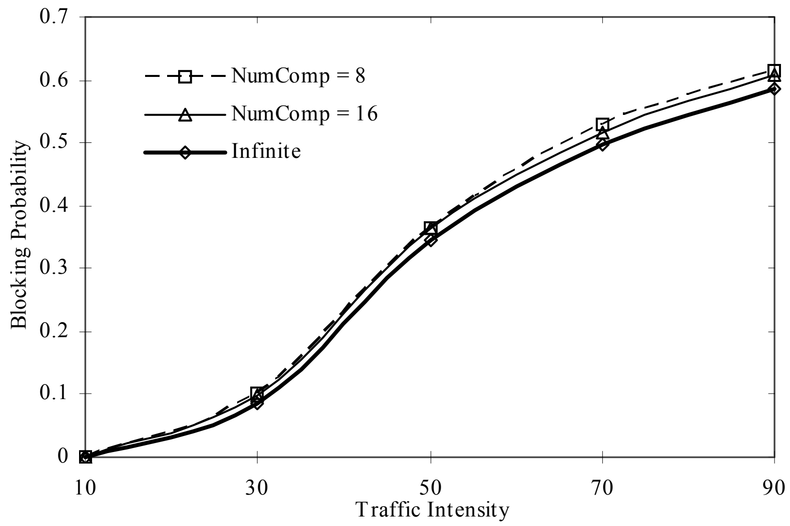


Figure 53 Blocking probability for various number of components

Figure 53 shows the relationship between blocking probability and number of components. Components include splitters and converters. Surprisingly, increase in number of components does not decrease in blocking probability significantly. The result is quite out of expectation. It is good news to save network hardware cost.

6.4 Conclusion

We have proposed an analytical model for blocking probability analysis on adaptive unicast routing over WDM networks with limited wavelength conversion in this chapter. Our model divides a route into segments, which differs from previous works. The segments are the basic units for analysis. Our contribution is to suggest an overflow model to get the blocking probability of a wavelength converter queue. Then, blocking probability and route selection probability are obtained using our formula. Finally, an iterative procedure is proposed to get the final network-wide blocking probability.

However, the analytical results of blocking performance are not close to the simulation results. This may be due to the imprecise assumptions and approximations that we have adopted for the sake of simplicity. First, peaked overflow traffic is approximated to be Poisson, which is generally not. Second, wavelength assignment to a segment is not strictly random. In fact, the wavelength of the current segment depends on the wavelengths of its previous segments. Third, the first and last segments of a route may not be complete. That further violates the random wavelength assignment assumption. Finally, the biggest shortcoming might be that we have not considered the conditional probability introduced by the sharing of common links among different connections. However, this is common to the existing works.

Our simulation results have revealed that number of wavelengths is the dominant factor in blocking performance, followed by density of wavelength conversion. The number of wavelength converters that a wavelength convertible node is equipped with has only limited impact on blocking performance. This observation may be used to guide network design.

We have also proposed an analytical model for blocking probability analysis on fixed multicast routing over WDM networks with limited capabilities. We divide the whole problem into two parts: blocking on links and on nodes. The nodes are classified into two types and each type is modeled using a queuing model. The Erlang loss formula is intensively used to calculate blocking probability on links and nodes. Finally, an iterative procedure is proposed to get the final network-wide blocking probability.

Despite its shortcomings, our analytical model provides a fast and easy way to approximate the blocking performance of a network without complex simulations. We will adopt new mathematical tools and improve the blocking performance analysis in future. Some existing works utilize the linear programming technique to calculate the blocking performance of WDM networks [36].

Chapter 7 Delay-Constrained Multicast over WDM Networks

7.1 Introduction

While the bandwidth of optical networks may be abundant, it may not be adequate to accommodate many multimedia applications. Hence, it is useful to study the delay-constrained multicast problem over WDM networks.

Table 8 Taxonomy of multicast algorithms

<table>
<tr><th></th><th>Offline multicast</th><th colspan="2">Online multicast</th></tr>
<tr><td>Unconstrained</td><td>Steiner tree
Shortest path tree
Core-based tree</td><td colspan="2">Shortest path tree
Core-based tree</td></tr>
<tr><td rowspan="2">Delay-constrained</td><td rowspan="2">NAÏVE [33]
2TREE_K [49]
2TREE_S [67]
CST_{CD}, CST_{C} [81]
CST_{P} [85]
CAO [70]</td><td>Non-rearrangeable</td><td>GREEDY [86]
CST_D [55]
WAVE [27]
BSMA [68]
SPH [74]</td></tr>
<tr><td>Rearrangeable</td><td>CDCRM [75]</td></tr>
</table>

Multimedia multicast over the electronic network has been extensively studied in the literature. Delay is one constraint imposed by multimedia multicast; others include jitter, bandwidth, and so on. Table 8 gives the taxonomy of the existing multicast algorithms. A brief survey on delay-constrained multicast is given in the next section.

The taxonomy is based on delay constraint and membership variation. Membership is static in offline multicast, and dynamic in online multicast. The delay-constrained online multicast problem can be further divided into two sub-groups: rearrangeable and non-rearrangeable. The criterion is whether changes in the multicast tree (i.e., rearrangement) are allowed when members join or leave the group. If the changes are allowed, the algorithm is called rearrangeable. Otherwise, it is non-rearrangeable.

In the table, the Steiner tree algorithm is the first choice for the unconstrained offline multicast problem. The algorithm emphasizes minimizing network cost. Strictly speaking, the shortest path tree algorithm belongs to the Steiner tree algorithm [29]. However, the algorithm is easy to fit into a distributed environment, and it is used by many protocols. Therefore, we have listed the algorithm individually. The shortest path tree algorithm and the core-based tree algorithm can be used for both offline and online multicast. The core-based tree algorithm aims to tackle issues (such as security, resource sharing, etc.) other than reducing network cost. In other words, core-based trees are not optimal in terms of network cost. Usually, the core-based tree is for sparse-mode multicast while the shortest path tree is for dense-mode multicast. There is a wealth of textbooks and papers on the algorithms. We shall now turn our focus to delay-constrained multicast.

Currently, only a few works have touched on the problem of delay-constrained multicast over WDM networks [87][88]. However, the existing proposals do not consider wavelength conversion. Hence, routing is trivial as it has no difference from the electrical counterpart. The problem to be considered is wavelength assignment. We intend to shed more light on the problem by introducing wavelength conversion.

7.2 Review of Related Works

7.2.1 Definition, Notation and Formulation

The definitions and notations have been described in Section 4.2. There are some algorithms for finding the delay-constrained least-cost path (DCP). The constrained Floyd (CF) algorithm adopts the dynamic programming technique and finds all DCPs between every pair of nodes [81]. The constrained Bellman-Ford (CBF) algorithm also adopts the dynamic programming technique and finds DCPs from one node to a set of nodes. The Blokh-Gutin (BG) algorithm is based on the Lagrangian relaxation method and is able to find the DCP between a pair of nodes. Note that it is not polynomial in general, but runs fairly fast in practice.

The delay-constrained multicast problem can be formulated as follows:

Given: **G**, s, **D** and a delay constraint Δ

Find: A tree **T** spanning **K**, such that:

- $\sum_{e \in T} c(e)$ is minimized
- $\sum_{e \in P(s,v)} d(e) < \Delta, \forall v \in D$

The problem is also known as the Constrained Steiner Tree (CST) problem. Note that we only consider uniform delay constraint, that is, all the destinations must satisfy the same delay constraint. Individual delay constraints can be solved by appropriate modifications to the corresponding algorithms.

In the case of dynamic multicast, the i^{th} multicast request r_i is denoted as (d_i, Θ), where $\Theta \in$ {join, leave} is the action taken. Accordingly, the multicast tree constructed is denoted as T_i.

7.2.2 Delay-Constrained Offline Multicast Algorithms

Naïve approach

The algorithm proposed in [33] aims to address the delay- and jitter-bounded multicast tree problem. Compared with normal algorithms, it deals with an extra constraint – jitter (i.e., delay variation). Here, we only describe the first phase that deals with delay constraint. The algorithm is quite simple. It constructs a shortest path tree in terms of delay. The authors also discuss how to apply the algorithm in the online multicast situation.

Two-tree approach

The two-tree approach constructs an MC-tree first. Then, the delay of the path from the source to each destination is checked. If it is violated, then the path is replaced by the path to the same destination in the DC-tree. To the best of our knowledge, no existing algorithm constructs a DC-tree first and then tries to minimize the cost by the replacement of paths in the MC-tree.

2TREE_K algorithm: In [49], the authors only consider a network with single metric cost. However, we can view it as a special case of the two-metric network where $c = d$ (i.e., $\gamma = 1$). The algorithm firstly constructs two trees: one MC-tree (by using an MST-based algorithm, e.g., Distance Network Heuristic [29]) and one DC-tree (by using any shortest path (in terms of delay) algorithm, called MINDC in [49]). Then each destination is examined to find the one whose cost difference in two trees is the biggest. Next, the path in the MC-tree is replaced with the path in the DC-tree. The replacement is repeated several times, and finally, the resultant network is pruned into a tree.

2TREE_S algorithm: In [67], the network is assumed to have two distinct metrics: delay and cost. The algorithm first constructs an MC-tree. Then the destinations that violate the delay constraint are removed. If there are destinations unconnected, a DC-tree that spans only these destinations are constructed. Finally, the two trees are merged and pruned appropriately. A similar routing algorithm is also used in [87] and [88]. The authors also deal with wavelength assignment, which is for the WDM network.

Direct approach

With the direct approach, the delay-constrained multicast tree is constructed directly by using DCPs. DCPs can be calculated using a few algorithms (cf. Section A.1.2).

$\mathbf{CST_{CD}}$ and $\mathbf{CST_{C}}$ algorithms: In [81], the dynamic programming approach is adopted to compute all DCPs between every pair of nodes. Thus, a closure graph G' (it is also a complete graph) is constructed by those DCPs. Then, a spanning tree T rooted at the sender is constructed until all receivers are connected. When considering whether to include some edge adjacent to node *v* in the spanning tree, two selection functions are defined:

$$f_{CD}(v,w) = \begin{cases} \dfrac{c(e(v,w))}{\Delta - (d(P_T(s,v)) + d(e(v,w))} & \text{if } d(P_T(s,v) + d(e(v,w)) < \Delta \\ \infty & \text{otherwise} \end{cases}$$

$$f_C(v,w) = \begin{cases} c(e(v,w)) & \text{if } d(P_T(s,v)) + d(e(v,w)) < \Delta \\ \infty & \text{otherwise} \end{cases}$$

The algorithms that use either f_{CD} or f_C are called CST_{CD} and CST_C, respectively. The final step is to replace the edge in T with the corresponding path in G, and prune to make a real tree.

$\mathbf{CST_P}$ algorithm: A similar algorithm is proposed in [85]. The closure graph is constructed in the same way as the above algorithms. However, the algorithm does not involve any selection function during MST construction. Instead, a modified Prim's algorithm [44] is adopted. The main idea is that whenever a new node is added, the existing on-tree nodes' linkage to their parents must be re-examined to guarantee delay constraint. Therefore, the tree is constantly changing before it is finally constructed.

CAO algorithm: In [70], the constrained Bellman-Ford algorithm is proposed to calculate DCPs from a node to a set of nodes. The authors propose a Constrained Adaptive Ordering (CAO) algorithm to address the CST problem. At every step, the CBF algorithm is run, and the shortest DCP is chosen between the source and one off-tree destination. Then, a merge algorithm is run to prune the redundant branches to get a tree shape. The algorithm runs till the tree is finally constructed. The time complexity is thus O(|K| * O('CBF')).

7.3 Problem Investigation

7.3.1 Formal Definition

The problem of delay-constrained multicast over the WDM network can be formulated as follows:

Given: A WDM network **G** with a set of wavelengths Γ, source *s*, destination set **D**, and delay constraint Δ

Find: A tree **T** spanning *s* and **D**, such that:

- $\sum_{e \in T} c(e)$ is minimized
- $\sum_{e \in P(s,v)} d(e) < \Delta, \forall v \in D$
- Each edge resides on a free wavelength. The allocation of wavelengths should not violate wavelength conversion capability distribution.

Note that the first two conditions do not include the cost and delay of wavelength conversion and light splitting (if any). This is solely for simplification of calculation.

The formal problem definition is very similar to the one given in Section 7.2.1. The biggest difference lies in the wavelength assignment sub-problem (i.e., the last condition). The last condition invalidates the direct use of the existing algorithms. The reasons are explained below.

7.3.2 Importance of Graph Transformation

Generally, the existing algorithms cannot be applied to the *topological graph* (Figure 54a, where the solid-gray node is wavelength inconvertible). There is no problem with the routing part, but the wavelength assignment part may fail. This is because of the existence of wavelength inconvertible (WIC) node(s).

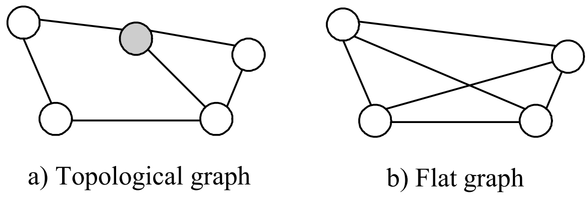


Figure 54 Illustration of different graphs

Suppose one branch of a multicast tree contains a WIC node (Figure 55a), and wavelength usage is as shown in Figure 55b (the dashed lines indicate occupied wavelengths). In this case, it is obvious that the wavelength assignment procedure fails at the WIC node.

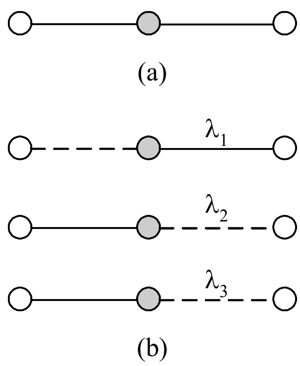


Figure 55 Illustration of one drawback of the topological graph

The *flat graph* (Figure 54b) looks similar to the topological graph, except that it does not contain any WIC nodes. Here, wavelength assignment will definitely succeed. However, unless necessary information is stored during transformation, there is no way to transform the flat graph back to the topological graph.

During transformation from the topological graph to the flat graph, network topology changes a little. To avoid dramatic change, we restrict the networks to be studied to have only a very small portion of WIC nodes.

7.3.3 Network Model

Only offline multicast will be studied. The networks studied will be restricted to networks with dense wavelength conversion. In other words, nearly all the nodes have wavelength conversion capability. Moreover, all nodes are assumed capable of light splitting.

For the sake of simplicity, each wavelength convertible (WC) node is assumed to have infinite wavelength conversion capability, and each light splitting capable node, infinite light splitting capability.

Certainly, nodes with finite capabilities can also be dealt with using DCSP and other similar techniques as described in the previous chapters. Considering finite capabilities will only make the problem slightly more difficult.

7.4 Proposed Algorithm

7.4.1 Graph Transformation

Transformation of WIC region

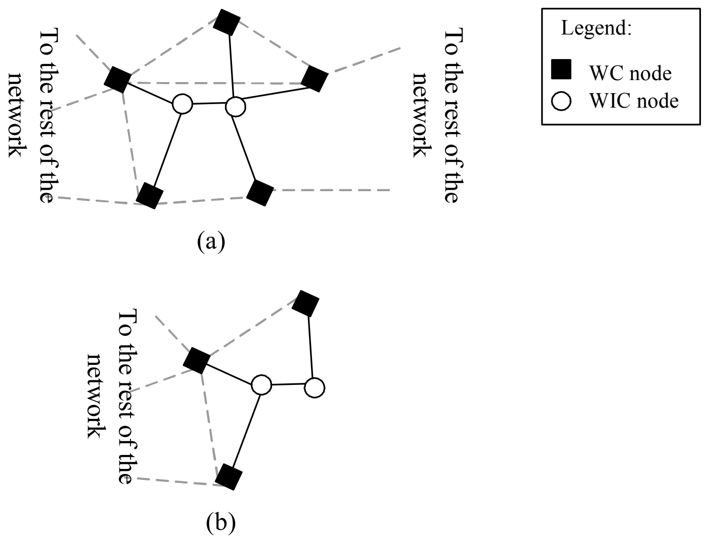


Figure 56 Illustration of the WIC region

The key concept is the *WIC region*. With every WIC node v, we associate a WIC region *W(v)* which is defined as follows: *W(v)* is the maximal connected part of the network that contains only WIC nodes as its interior nodes. The exterior nodes of the WIC-region are either the WC nodes or the outermost WIC nodes. Figure 56 illustrates two WIC regions. The WIC region comprises WIC and WC nodes, and the solid lines that connect them.

Since the regions outside the WIC region(s) do not break the wavelength assignment process, it is the WIC region that needs to be transformed and made wavelength assignment friendly.

For simplicity, suppose the WDM network has only two wavelengths. Taking the WIC region in Figure 56a as an example, the transformation process is illustrated in Figure 57.

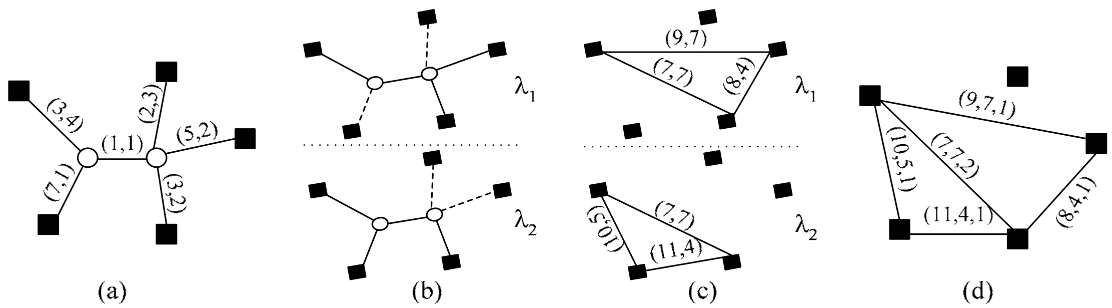


Figure 57 Illustration of WIC region transformation

Figure 57a is a duplication of Figure 56a, except that every edge is associated with a (*c*, *d*) tuple. In the tuple, *c* stands for cost and *d* stands for delay. Figure 57b shows utilization on each wavelength. The dashed line indicates that the corresponding wavelength on the edge is occupied. Figure 57c depicts the virtual topology on each wavelength after the WIC nodes are removed. As WIC nodes are rare, it is simple to generate the virtual topology.

Finally, the virtual topologies need to be merged to generate one flat topology (Figure 57d). Note that edges in the WIC region are actually short paths. Thus, they are called virtual edges. If there is only one virtual edge between two WC nodes, merge operation is quite straightforward. If there is more than one virtual edge with the same (c, d) tuple, it is worth recording the number of such edges. Thus, each edge now is associated with a triplet (c, d, m), where m stands for the number of repeats. The most complex case is where there is more than one virtual edge with different tuples. In this case, the perfect candidate should have minimal cost, minimal delay and maximal repeat. One possible formula may be:

$$f(e) = \frac{m}{c \times d}, \forall \text{ edge } e \text{ in WIC region} \quad (1)$$

The virtual edge with maximum f-value wins. Of course, there are other selection formulae for special needs.

Merging with non-wic region

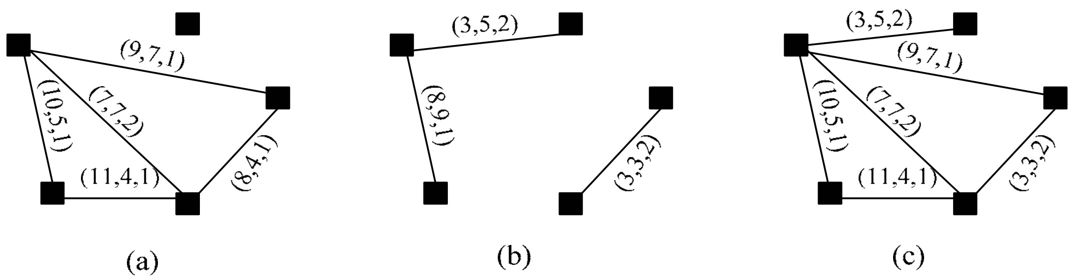


Figure 58 Merge with non-WIC region

The tricky part occurs when exterior WC nodes have direct edges connected in the non-WIC region. Figure 58 depicts such an example. Figure 58a is a duplication of Figure 57d, which shows a WIC region. Figure 58b shows the part of the non-WIC region that contains only the exterior WC nodes of the WIC region. The m element in the triplet stands for the number of free wavelengths on the edge. It is obvious that there are two pairs of nodes that have more than one edge. Thus, the above formula (1) is used here to decide which edge is to stay and which to leave. The final result is shown in Figure 58c.

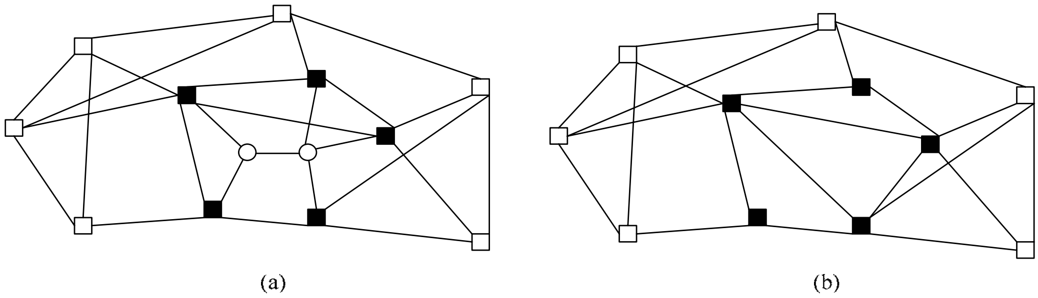


Figure 59 An example of the flat graph

Continuing with the previous examples, Figure 59a shows the original WDM network with two wavelengths. The solid square nodes and the round nodes form the WIC region in Figure 56a. The square nodes are all wavelength convertible. Figure 59b is the transformed flat graph.

Membership transfer

If any WIC node happens to be a member of the multicast group, the membership will be taken over by the WC node nearest it. In case there is more than one such node, one will be arbitrarily chosen. Another possible way is to transfer the membership to all the adjacent WC nodes. However, this option is complex as the membership of those WC nodes is exclusive.

Now, the WIC nodes are removed from the graph, the membership is transferred, and the graph is flattened. Due to the fact that every node has ample wavelength conversion capability, the multicast routing and wavelength assignment problems can be divided and considered separately. It should be emphasized that the routing result can certainly be matched with an appropriate wavelength assignment scheme.

7.4.2 Formal Algorithm Description

Most of the algorithms described in Section 7.2 can be applied to the flat graph with no or little modification. After routing is done, the next step is to do wavelength assignment. There are many existing algorithms for that (cf. Section 2.4.3). The easiest way here is to randomly pick a wavelength in the candidate set.

Normal algorithms do not consider the third element m. However, proper use of the element may improve performance. Therefore, we propose several approaches which consider that element when doing routing.

First, for the sake of high blocking performance, it is beneficial to protect the hot edges. In our case, if m is less than a certain threshold, the edge will be temporarily unavailable for routing. Those edges will only be available if no solution can be found without them.

Second, a selection formula based on the triple can be defined, such as formula (1) in the previous sub-section. Then, when routing, the edge with higher value has a higher priority to be included in the final solution. This approach best suits the distributed routing environment.

Main flow

1. Find the WIC regions and flatten them. Do membership transfer whenever necessary
2. Merge all the regions into one, and obtain the flat graph
3. Do multicast routing on the flat graph
4. Assign wavelengths to the tree
5. Replace the virtual edges in the tree, and do the necessary pruning

Figure 60 Pseudo-code of the main flow

Figure 60 lists the main algorithm flow. The algorithm is self-explanatory. Steps 1 and 2 deal with graph transformation. The graph transformation method has been elaborated on in the previous sub-section. Step 3 finishes multicast routing, and Step 4 assigns the wavelengths.

Multicast routing

```
while (destinations are left unconnected) {
    Find the nearest destination to the current tree in terms of cost
    Calculate a DCP between the two endpoints
    Connect the destination to the tree through the DCP
}
```

Figure 61 A multicast routing algorithm

Generally, the existing multicast routing algorithms can be used. Figure 61 lists our algorithm. It is similar to the SPH heuristic [29]. The only difference is that we use DCP instead of LCP to connect the destination to the tree.

Wavelength assignment

There are many wavelength assignment algorithms. Most of them can be used with appropriate modifications. Here, we propose another algorithm that maximizes the use of light splitting to speed up packet forwarding. The algorithm is described as in Figure 62.

Take Figure 63 as a sample multicast tree. Every edge is attached to a set of available wavelengths and is in *initial* state. There are two other states, namely, *assigned* and *processing*. The tree is organized so that it is rooted at the source and the other nodes are ordered in levels according to their distance to the source. A depth-first traversal is performed during wavelength assignment. The key is to calculate the intersection of sets of related edges. The set of thick edges will intersect with its right-side siblings (its left-side siblings are guaranteed to be assigned due to depth-first traversal) and its parent (if it is not assigned). The related edges are included in the dashed-line irregular polygon.

```
level = 1; Set all edges in initial state;
while (having edges not assigned at the current level) {
  Locate the leftmost initial edge, e at the current level;
  if (none can be located) {
    level++;
    continue;
  }
  Perform intersection of sets among edge e and its unassigned parent and siblings;
  if (intersection result is empty) { // move back to source and assign wavelength
    Choose any one wavelength from any processing edge;
    Assign the wavelength to all processing edges and set their state to assigned;
    Set edge e to processing state;
    level++;
    Process e's child edges;
  } else {
    Replace the sets of those edges with intersection and set their state to processing;
    if (e has children) {
      level++;
      Process e's child edges;
    } else
      level = 1; // back to the first level
  }
}
```

Figure 62 Wavelength assignment algorithm that makes use of light splitting capability

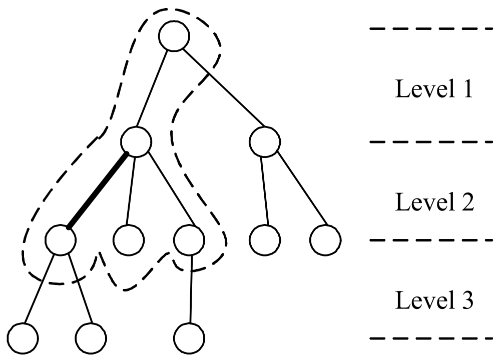


Figure 63 Sample multicast tree

After wavelength assignment, the last step is to restore virtual edges in the tree and prune the resultant tree. The procedure is easy and we will not describe it here.

7.5 Simulation

7.5.1 Simulation Setup

Network topology

Table 9 Parameters of simulations

N	30
L	60
α	0.4
β	0.3
numWave	{4, 8, 12, 16}
tDense	{10%, 20%, 30%}
lDense	100%
wDense	{40%, 60%, 80%, 100%}
ρ	{10, 20, 30, 40}
ξ	{0.8, 1.0, 1.2}

The Waxman model is adopted to generate random networks with WAN characteristics. The parameters in the simulations are set from the value fields listed in

Table 9. Due to the infinite capabilities assumption, the numLS, Lscap and numWC parameters are set to be infinite. The networks generated have 30 nodes and an average degree around 4. They resemble the wide-area network well. The value fields of the parameters are given in Table 9. Each combination of parameters will generate a few instances, and each instance will run sufficiently long to obtain blocking probability.

The delay is decided as follows. Suppose network diameter in terms of delay is x when the request arrives; then the delay constraint for the request will be ξx. The delay constraint is very tight when ξ=0.8, and it is relaxed when ξ increases.

DCP calculation

Any algorithm in Section 7.2 can be used for DCP calculation. In our simulation, we adopt the Blokh-Gutin Algorithm. Our simulation shows that the algorithm runs fairly fast and efficiently.

7.5.2 Numerical Results

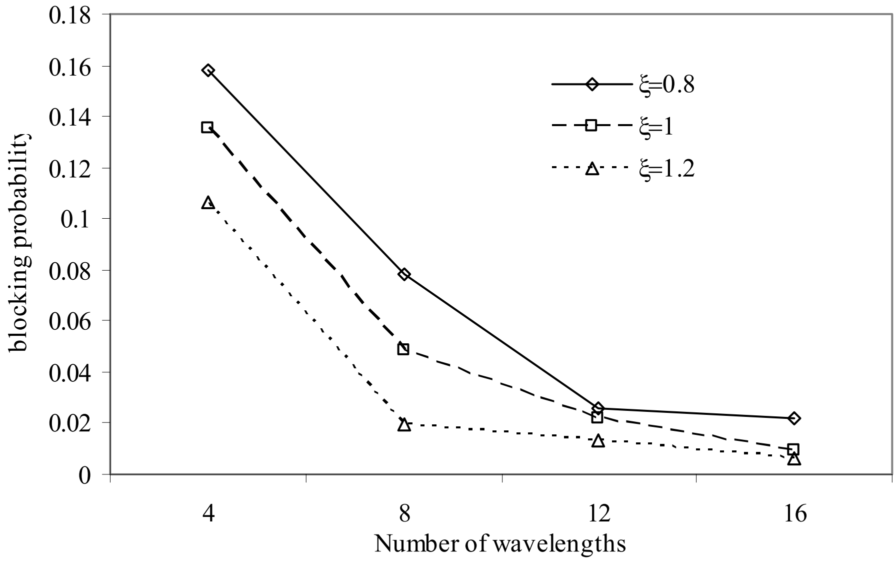


Figure 64 Blocking probability vs. number of wavelengths

On the whole, the algorithm runs fairly fast. Almost all requests can be entertained within 30 milliseconds. As we have stated a number of times, we view blocking performance as the most important metric in a wavelength-routed network.

Figure 64 shows the blocking probability variation with the number of wavelengths for different delay coefficients. Not surprisingly, blocking probability drops with the increase in number of wavelengths. In other words, blocking performance improves. It is noticeable that performance improvement saturates when the number is greater than 12. This implies that 12 wavelengths are enough for networks of our type when traffic intensity is less than 40. The following results confirm this statement.

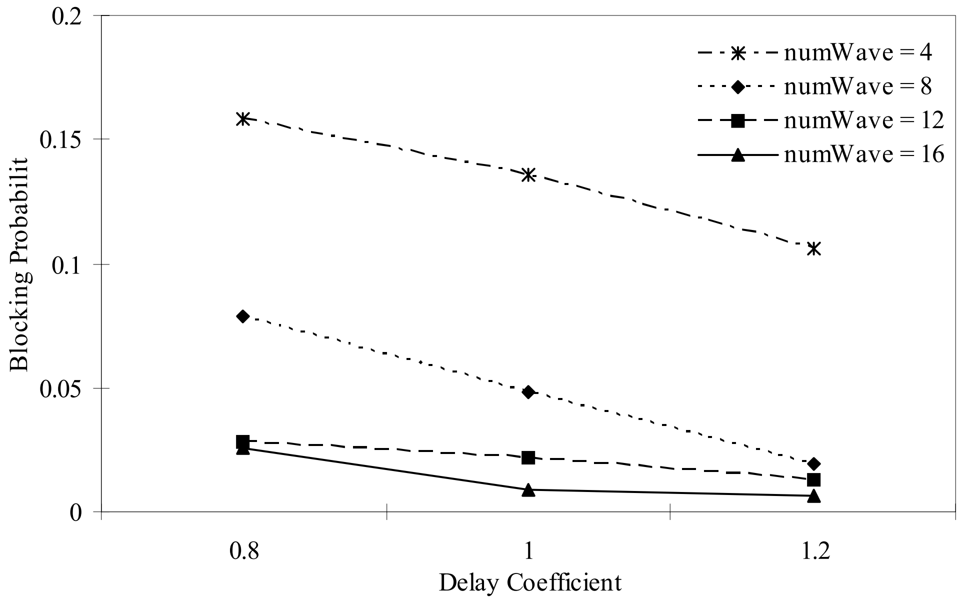


Figure 65 Blocking probability vs. delay coefficient

Figure 65 shows blocking probability against the delay coefficient. With the easing of delay constraint, blocking performance improves steadily. Again, the saturation effect sets in.

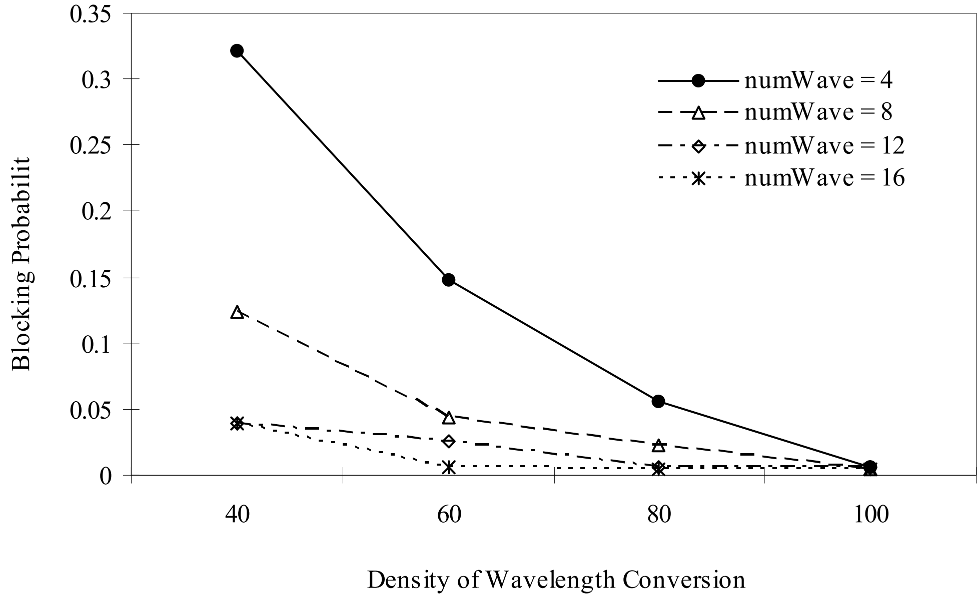


Figure 66 Blocking probability vs. density of wavelength conversion (ξ = 0.8)

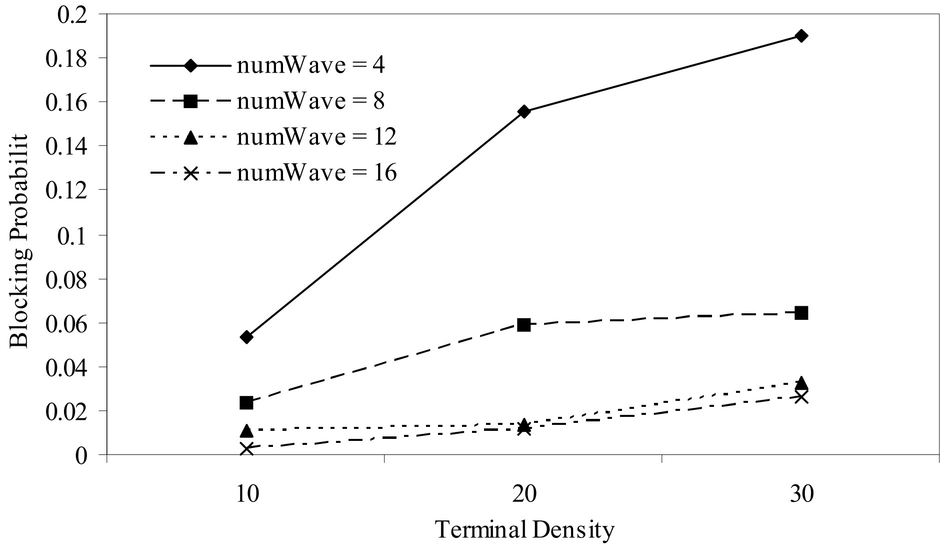


Figure 67 Blocking probability vs. terminal density (ξ = 0.8)

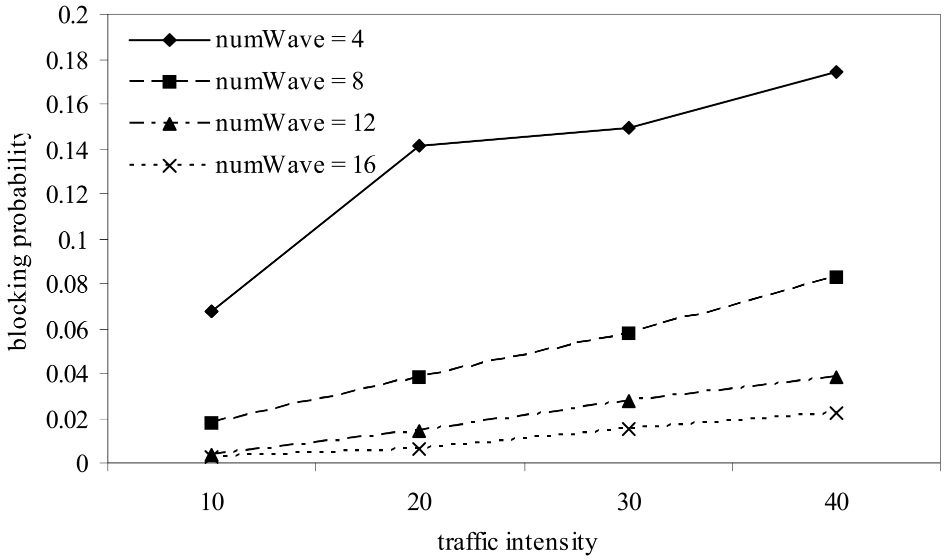


Figure 68 Blocking probability vs. traffic intensity (ξ = 0.8)

Figure 66 to Figure 68 show the relationship between blocking probability and other factors when $\xi = 0.8$. Figure 66 confirms our initial assumption about dense wavelength conversion. When the density of wavelength conversion is less than 60%, blocking performance deteriorates heavily, especially when the number of wavelengths is small. Additionally, saturation of the number of wavelengths is clear in the figure.

Figure 67 depicts blocking probability against terminal density. Not surprisingly, blocking performance degrades significantly when terminal density increases. But when the number of wavelengths is big, the degradation of blocking performance is not significant.

Figure 68 illustrates blocking probability variation with traffic intensity. Basically, blocking probability increases linearly with traffic intensity. The benefit that is brought by the number of wavelengths increases faster than the increase in number of wavelengths itself. The increase slows down when the number of wavelengths is more than 12. Again, this is the saturation effect.

7.6 Conclusion

Delay-constrained multicast is part of multimedia multicast. However, multimedia multicast includes many other constraints such as jitter, bandwidth and so on. Delay-constrained multicast over the electronic network has been extensively studied, and we have given a brief survey.

On the other hand, delay-constrained multicast is a new topic in WDM network research. The case of the dense wavelength conversion network is studied. The key concept is the WIC region. By introducing this concept, wavelength inconvertible nodes can be removed, and the whole network topology can be transformed to a flat graph. After transformation, all existing algorithms can be easily applied with almost no change. The existing abundant algorithms can be used for wavelength assignment. In addition, we have proposed an algorithm that makes extensive use of light splitting.

The concept of the WIC region cannot be applied to solve the same problem in the sparse wavelength conversion network because graph transformation will heavily change the topology, and blocking performance would then drop severely.

Chapter 8 Conclusion and Future Work

8.1 Problem Revisited

In this thesis, we have discussed the problem of multicast over WDM networks. WDM networks can be divided into a few categories (Figure 1). Studies have been made in each category, and we have reviewed them in Chapter 2. Our focus is on the wavelength-routed network, which adopts the mesh topology and uses the circuit-switching technique. Compared with traditional IP multicast, multicast over WDM networks entails the additional wavelength assignment problem. Therefore, it is usually referred to as the multicast routing and wavelength assignment (MCRWA) problem.

Numerous research works have been done in this field [24][36][40][41][47][52][61][62][84][89][90]. However, they all suffer from some shortcomings. First, they tackle the two sub-problems of multicast routing and wavelength assignment individually. Second, they all assume that the target networks have infinite light splitting and wavelength conversion capabilities. Hence, the results of these studies lack practical significance. Third, some proposals try to solve the multicast routing problem at the IP layer, and then map the solution down to the WDM layer. However, these proposals inevitably involve the re-routing procedure, which causes turbulence in transmission and uncertainty in routing. Finally, some proposals suggest the construction of a backbone sub-network among some powerful nodes in advance, and then connect the rest of the members to the backbone. However, that approach creates bottlenecks in the backbone.

8.2 Thesis Contributions

We have overcome the shortcomings listed above with our methods that can deal with wavelength-routed networks with finite capabilities. Our methods can also complete routing and wavelength assignment in one phase. Additionally, our methods dynamically route requests and adapt to changes in network utilization. Our methods use the degree-constrained Steiner problem to model finite light splitting capability.

We have developed the expanded-graph model to transform a WDM network into its corresponding graph. The model greatly simplifies the resultant graph, compared with the auxiliary-graph model [50]. As the result of our model is a graph, Graph Theory algorithms can be employed with only minor changes.

To further reduce computational complexity and yet maintain quality of solution, we have created the layered-routing approach. The approach corresponds to the multi-layer nature of the WDM network. Based on the approach, we have designed two generic algorithm frameworks to guide the transplantation of Graph Theory algorithms.

We have compared our two methods – the expanded-graph model and the layered-routing approach – through extensive simulations. As expected, the expanded-graph model outperforms the layered-routing approach at the cost of much higher computational complexity. The choice between the two methods is a tradeoff between computational complexity and quality of solution.

We have also carried out some additional theoretical analysis. Queuing network is applied to analyze blocking performance and to cover more situations. Our analytical models can deal with limited wavelength conversion and light splitting capabilities at a node. Compared with other theoretical analysis of blocking performance, the networks that we have used are much closer to real situations, and much more difficult to analyze.

In conclusion, only a few research works have been done from the aspect of Graph Theory to tackle the MCRWA problem. We have provided an efficient model to facilitate graph transformation from WDM networks and provide high-quality solution with reasonable time complexity. We have introduced the layered-routing approach for satisfactory results in fairly fast speed. Most importantly, ours is the first study where real WDM networks with finite light splitting and wavelength conversion are considered. It is also the first time that the delay-constrained Steiner problem is introduced into WDM networks.

8.3 Future Work

8.3.1 Group Steiner Tree Problem

A simpler representation of the WDM network can be formed on the Group Steiner problem [20]. Every node in the WDM network is a group of nodes. Figure 69 shows the transformation. Figure 69a shows the original WDM network, and Figure 69b is the Group Steiner representation. All the small circles inside a big circle form a group. Here, we assume the WDM network has four wavelengths, so every group has four elements.

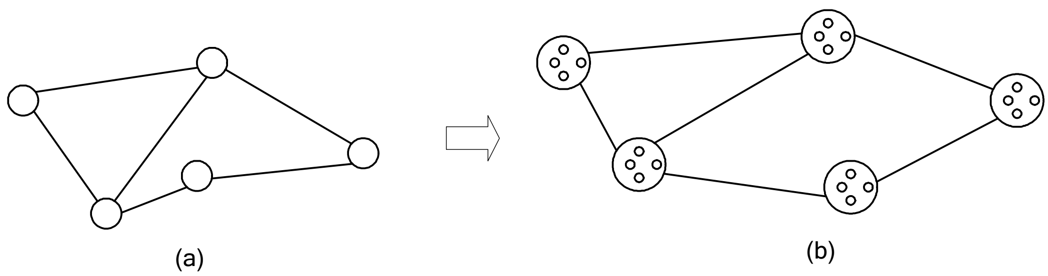


Figure 69 Group Steiner representation of the WDM network

The representation seems much simpler. However, the Group Steiner problem is itself NP-complete, and still under research. Nevertheless, it is a possible area of further study.

8.3.2 Multicast over Packet-Switched WDM Network

The full packet-switched network will be the final target of the WDM network evolution. However, many restrictions hinder the evolution. For instance, optical RAM and quantum computing are critical to the full packet-switched network. As stated before, the upper bound of optical transmission is 40 Gbps if O/E/O conversion is involved [71]. Therefore, packets must be stored in the optical domain and randomly accessed by interim switches. Currently, the fiber delay line (FDL) can be used as optical memory, but it lacks the ability of being randomly accessible. Quantum computing is impossible to current technology. Without these two techniques, packet-switched WDM networks are impossible.

However, some pioneering works have been done on current partial packet-switched networks, such as the OLS and OBS networks [35][58][59][92]. The optical Internet architecture is also a good reference [25][52][56][64]. Moreover, new packet-switched techniques can be proposed. In short, this is an open field.

8.3.3 Delay-Constrained MCRWA

Much work has been done on multicast over WDM networks. A new field is the delay-constrained MCRWA problem. Only a few research works [88] in this field exist, and none of them considers wavelength conversion. Our preliminary results on delay-constrained multicast over WDM networks with dense wavelength conversion are presented in Chapter 7.

For extra complexity, other constraints, such as jitter and bandwidth, may be added as well. In a broader sense, QoS multicast over WDM networks is another challenge.

8.3.4 Online Multicast

Offline multicast requires full knowledge of group members before routing. This is to say: the group is fixed throughout the session. This may be insufficient for a multicast conference, where members may freely join and leave. It is an interesting and yet difficult problem to support online multicast. To the best of our knowledge, no existing works deal with online multicast over wavelength-routed networks. The difficulties are: 1) to minimize topology change and service disturbance if rearrangement is allowed, 2) to maximize blocking performance, which requires careful allocation of wavelengths and links, and 3) to minimize response time. As a tentative first step toward a solution, Chapter 7 provides some existing algorithms on online multicast, especially on node addition, node removal, and rearrangement.

Appendix A. WDM Technology

Optical fiber communication is now firmly established as the preferred means of communication for signals over a few hundred megabits per second over distances more than a few hundred meters [71]. In the early days, fibers were merely used as a reliable, fast, and secure substitute of transmission media for the coppers. Restricted by technologies at that time, only one channel is carried over the fiber. The first commercial development and deployment of WDM in optical fiber transmission systems occurred in the early 1990s at Bell Labs [43]. WDM stands for Wavelength Division Multiplexing, which is a kind of technology that provides many communication channels at various wavelengths. Thus, the capacity of a single fiber is increased to many folds.

A.1. Brief History of WDM Network

The emergence and improvement of the WDM technology is not the consequence of any single technology. It is a result of interaction among many techniques. The list surely includes the optical fiber. In addition, other techniques, such as wavelength division multiplexing, optical crossconnect, are also included in the list.

A.1.1. Evolution of Optical Fiber

Figure 70 shows the absorption characteristics (attenuation) of different fibers. The upper curve shows the absorption characteristics of early (in the 1970s) fibers. The lower one is for modern fibers. With the improvement of fiber technology, the attenuation of fiber is no longer the major factor that affects the transmission quality. Figure 70 also shows the three transmission windows (wavelength band) in the transmission spectrum. The wavelength band used by a system is an extremely important defining characteristic of that optical system [37].

The Short Wave Band is around 800-900 nm. This was firstly used for optical communication in the 1970s and early 1980s. It was attractive because 1) the attenuation is relatively low (refer to the upper curve in Figure 70, where the dashed-line part is not precise and shows only the trend), and 2) the low-cost optical sources and detectors can be used.

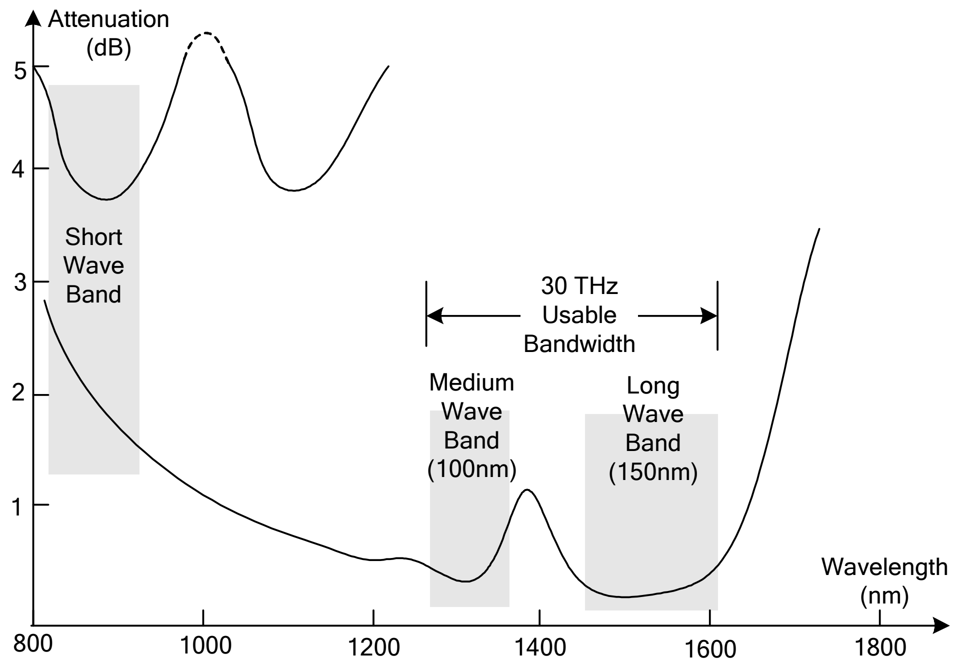


Figure 70 Transmission windows and attenuation characteristics

The Medium Window Band is around 1310nm (about 100 nm wide), which came into use in the mid 1980s. Though the sources and detectors working in this band are more costly, the majority of today's long-distance communications systems operate in it, for its low fiber attenuation and near-zero dispersion (for single-mode fibers).

The Long Wave Band is around 1550 nm (about 150 nm wide). The fiber attenuation is extremely low for current fiber, and current commercially available optical amplifiers (e.g. Erbium-doped fiber amplifier, EDFA) are also working in this band. However, the dispersion is relatively high. There are a few ways to compensate the dispersion, such as to use the Large Effective Area Fiber (LEAF) introduced by the

Corning Company. In the late 1990s, almost all new communications systems operate in this band.

The total range of the medium and long wave bands is about 250 nm. By directly converting from wavelength to frequency, the potential bandwidth of the low-loss region is approximately 30 THz [16].

A.1.2. Evolution of Optical Fiber Transmission Systems

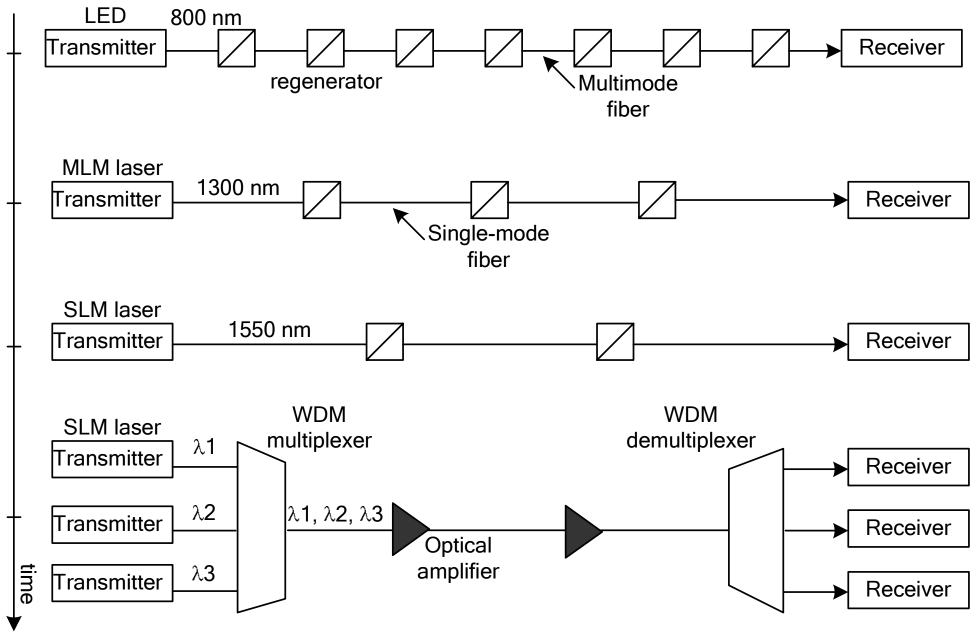


Figure 71 Evolution of optical fiber transmission systems

Figure 71 depicts the evolution of the transmission system. At the early days (1970s), the Light-Emitting Diodes (LEDs) are used for transmission over multimode fiber at 800 nm. Later, Multi-Longitudinal Mode (MLM) lasers are used and the transmission window is shifted to 1330 nm, because of the invention of the new optical fiber and its low attenuation waveband. Then, Single-Longitudinal Mode (SLM) lasers working on 1550 nm wavelength dominate the market. Information on the laser technology is to be discussed in the next sub-sections. Before the invention of optical amplifiers, the signals are regenerated at intermediate stations using regenerators. The regenerators require Optical/Electrical/Optical (O/E/O) conversion, and hence are bottlenecks of the whole system.

With the commercialization of the Erbium-doped fiber amplifier (EDFA) at 1550nm window (1990s), the signals can be directly amplified in the optical domain simultaneously. Only then, the Dense WDM (DWDM) technology becomes practical and feasible. ITU-T decided that the reference frequency of a DWDM system is 193.1THz (i.e., 1552.52nm). The gap between neighboring wavelengths should be integral multiple of 100GHz (approximately, 0.8nm). Recently, the systems that use 25 GHz spacing have appeared [72].

Note that, even before the EDFA, it is possible to multiplex signals and transmit through fibers. The problem is that only two wavelengths (i.e., 1310 nm and 1550 nm) can be multiplexed. This is called the Coarse WDM (CWDM) system, which provides full-duplex bidirectional communications with the 1310 nm for upstream, and the 1550 nm for downstream transmission [91]. From now on, we regard WDM and DWDM as synonymous, unless otherwise stated.

A.1.3. Evolution of Optical Networking

The network itself evolves from the single-span transmission to real optical networking. The systems shown in Figure 71 are no more than single-transmission systems. Figure 72 depicts the evolutionary course, beginning with single-channel point-to-point transmission systems and leading to optical networking. The discovery of EDFA and other amplifiers invented later made extension of network span possible. The invention of Wavelength Add/Drop Multiplexer (WADM) [18] and Optical Crossconnect (OXC) makes optical networking feasible.

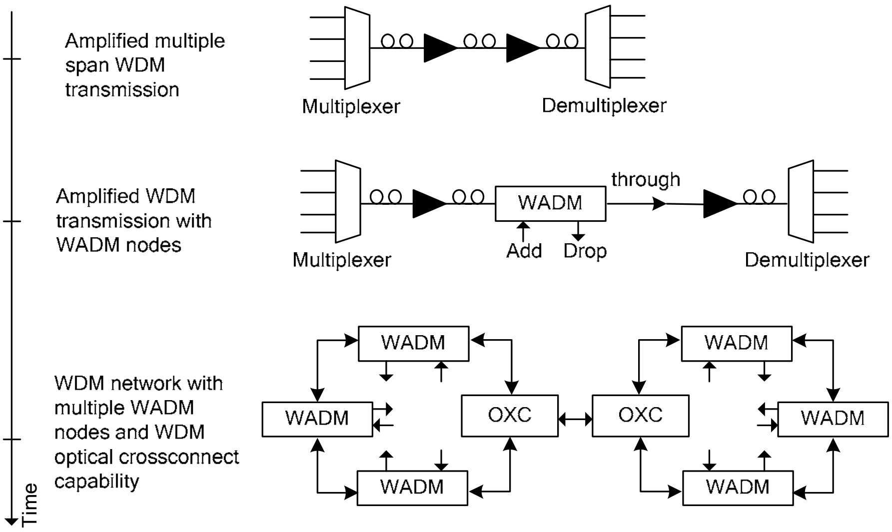


Figure 72 Evolution of fiber-optic transmission from single-span transmission to optical networking

There are three scenarios to upgrade existing fiber-based networks with WDM technology: (1) the existing fiber plant is used as such, (2) WDM is introduced as a point-to-point technology, and (3) WDM is introduced as a networking technology by adding an optical switch to each IP router.

WADMs facilitate the management of fiber capacity by enabling the selective removal and reinsertion of WDM channels at intermediate points in the line system. The consequences are tremendous fiber capacity and economical fiber utilization.

OXCs will play significant roles in future optical networks. Its functionality include reliable signal monitoring, fast restoration process at the optical layer, extensive bandwidth management, and traffic grooming. This is explained further in the next sub-section.

Most WADMs and OXCs have an optical switch as its core component. Actually, with the WADM becoming more and more powerful, the distinction between the WADM and OXC will become blur.

A.2. Optical Components

To build a WDM network requires many optical components. There are tons of optical components off the shelf. Hereby, we only briefly describe the two components that critical to optical multicast, that is, wavelength converter and light splitter. Interested readers are referred to [37] and [72], and references therein, for in-depth discussions and other components.

A.2.1. Wavelength Converter

Opto-electrical conversion is the choice of most current commercial systems. In some cases, such as the access network is still using LEDs on multi-mode fibers, all-optical conversion technology won't work, and O/E/O conversion is the only viable technology. Moreover, Current wavelength converters require an extra laser source as pump, which is somewhat an unnecessary expense (Figure 73). However, it is only a short-term interim technology, especially when the line speed increases beyond the limit of electrical threshold, 40 Gb/s [71]. Actually, 40 Gb/s transmission is currently a hot topic in industry.

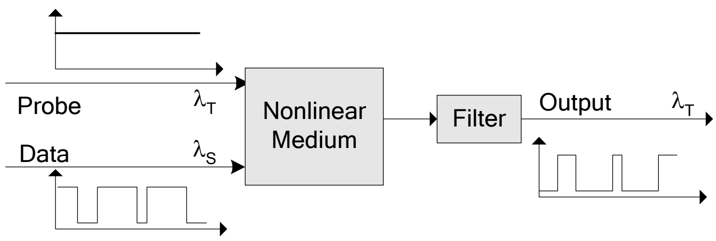


Figure 73 Principal model of wavelength converter

Generally, an all-optical wavelength converter can be viewed as a four-terminal device with three inputs and one output. The information-bearing signal at a wavelength λs, a continuous wave (CW) probe signal (which may or may not be at the target wavelength λ_T depending on conversion method), and an electronic control signal form the inputs. The output is a data-bearing signal (with or without logical bitstream inversion) at the target wavelength λ_T. Figure 73 shows the basic structure of a wavelength converter. The nonlinear media is where the conversion happens. It may be a third-order medium (e.g., standard fiber, Semiconductor Optical Amplifier (SOA), etc.), or a second-order medium (e.g., a lithium niobate waveguide), depending on the method used [37][80].

The network performance improvements offered by WCs depend on a number of factors, including network topology and size, the number of wavelengths, and the routing and wavelength assignment algorithms used. Limited wavelength conversion provides close performance to that achieved with ideal wavelength conversion, when the nodes are equipped with tunable transceivers [46].

A.2.2. Light Splitter

The basic principle behind light splitting is the resonant coupling. Resonant coupling has many special features, and is used to build various kinds of couplers and splitters.

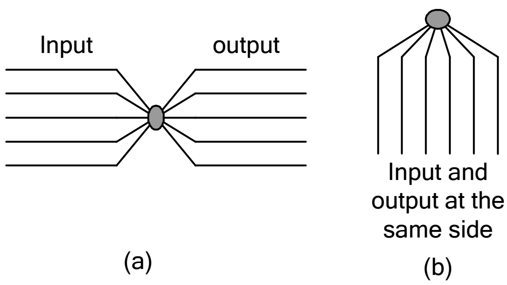


Figure 74 Star couplers

The principle is depicted in Figure 75. The power of light oscillates between two closely placed identical single-mode fiber cores. The level of darkness indicates the intensity of power. By properly choosing the length of fibers (coupling length), the incident light from port 4 can be equally split, and emitted from port 2 and 3. For a more theoretical explanation, refer to the textbooks on fiber-optics.

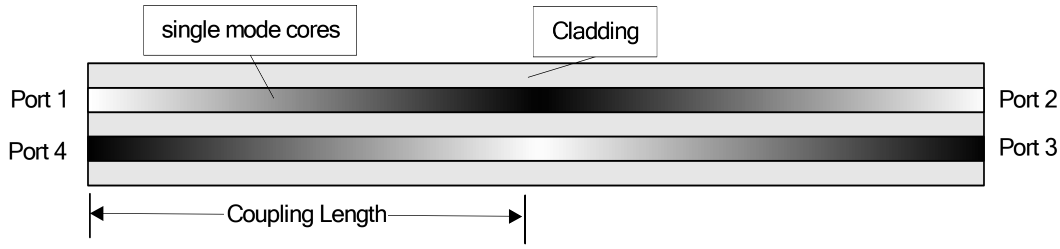


Figure 75 Principle of resonant coupling

If several fiber cores are melted together as a plate, this produces the star couplers (see Figure 74). The star coupler is key element in broadcast-and-select WDM network, a kind of single-hop network. The star coupler is also important part of a multicast-capable switch.

AUTHOR'S PUBLICATIONS

Journal Papers:

[1] Aijun Ding and Gee-Swee Poo, A Survey of Optical Multicast over WDM Networks, Computer Communications, vol. 26, no. 2, pp. 193-200, Feb. 2003

Conference Papers:

[2] Ajiun Ding, Gee-Swee Poo, and Sun-Teck Tan, An Expanded Graph Model for MCRWA Problem in WDM Networks, in proc. IEEE LCN, pp. 557-564, Nov. 2002

[3] Aijun Ding, Gee-Swee Poo, and Sun-Teck Tan, Multi-Layer Model for Optical Multicast over WDM Networks, in Proceedings of First International Conference on Optical Communications and Networks (ICOCN), pp. 274-277, Sept. 2002

[4] Aijun Ding, Sun-Teck Tan, and Gee-Swee Poo, Blocking Performance on Adaptive Routing over WDM Networks with Sparse Wavelength Conversion, in proc. IEEE LCN, pp. 187-193, Oct. 2003

Reference

[1] Ahmed Mokhtar, and Murat Azizoglu, *Adaptive Wavelength Routing in All-Optical Networks,* IEEE/ACM Transactions on Networking, vol. 6, no. 2, pp. 197-206, April 1998

[2] Ajiun Ding, Gee-Swee Poo, and Sun-Teck Tan, *An Expanded Graph Model for MCRWA Problem in WDM Networks*, in proc. IEEE LCN, pp. 557-564, 2002

[3] Aijun Ding and Gee-Swee Poo, *A Survey of Optical Multicast over WDM Networks*, Computer Communications, vol. 26, no. 2, pp. 193-200, Feb. 2003

[4] Aijun Ding, Gee-Swee Poo, and Sun-Teck Tan, *Multi-Layer Model for Optical Multicast over WDM Networks*, in Proceedings of First International Conference on Optical Communications and Networks (ICOCN), pp. 274-277, 2002

[5] Alastair M. Glass, David J. DiGiovani, et al., *Advances in Fiber Optics*, Bell Labs Technical Journal, pp. 168-187, January-March 2000

[6] Albert Greenberg, Gisli Hjalmtysson and Jennifer Yates, *Smart Routers – Simple Optics, A Network Architecture for IP over WDM*, Optical Fiber Communication Conference (OFC) 2000, vol. 3, pp. 292-294, 2000

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

[8] Andre Girard, *Routing and Dimensioning in Circuit-Switched Networks*, Addison-Wesley, 1994

[9] Ayan Banerjee, John Drake, et al., *Generalized Multipotocol Label Switching: An Overview of Routing and Management Enhancements*, IEEE Communications Magazine, pp. 144-150, Jan. 2001

[10] Ayan Banerjee, John Drake, et al., *Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques*, IEEE Communications Magazine, pp. 144-151, July 2001

[11] B. M. Waxman, *Routing of Multipoint Connections*, IEEE JSAC, vol. 6, no. 9, pp.1617-1622, Dec. 1988

[12] Bala Rajagopalan, James Luciani, et al., *IP over Optical Networks – A Framework*, Internet Draft, July 2001

[13] Bala Rajagopalan, Dimitrios Pendarakis, et al., *IP over Optical Networks: Architectural Aspects*, IEEE Communications Magazine, pp. 94-102, Sept. 2000

[14] Bernhard Korte, and Jens Vygen, *Combinatorial Optimization: Theory and Algorithms*, Springer, 2000

[15] Bharat T. Doshi, Subrahmanyam Dravida, et al., *A Simple Data Link Protocol for High-Speed Packet Networks*, Bell Labs Technical Journal, pp. 85-103, January-March 1999

[16] Biswanath Mukherjee, *WDM-Based Local Lightwave Networks, Part I: Single-Hop Systems*, IEEE Network, pp. 12-27, May 1992

[17] Biswanath Mukherjee, *WDM-Based Local Lightwave Networks, Part II: Multihop Systems*, IEEE Network, pp. 20-32, July 1992

[18] Byrav Ramamurthy and Biswanath Mukherjee, *Wavelength Conversion in WDM Networks*, IEEE Journal on Selected Areas in Communications, vol. 16, pp. 1061-1073, September 1996

[19] C. Randy Giles, and Magaly Spector, *The Wavelength Add/Drop Multiplexer for Lightwave Communication Networks*, Bell Labs Technical Journal, pp. 207-229, Jan.-March 1999

[20] C. S. Helvig, Gabriel Robins, and Alexander Zelikovsky, *Improved Approximation Bounds for the Group Steiner Problem*, in proc. conf. On Design Automation and Test in Europe (DATE'99), Paris, pp. 406-413, Feb. 1998

[21] Chien Chen and Subrata Banerjee, *A New Model for Optimal Routing and Wavelength Assignment in Wavelength Division Multiplexed Optical Networks*, Proceedings of IEEE INFOCOM '96, pp. 164-171, 1996

[22] Ching-Fang Hsu, Te-Lung Liu, and Nen-Fu Huang, *Performance of Adaptive Routing Strategies in Wavelength-Routed Networks*, IEEE International Conference on Performance, Computing and Communications, pp. 163-170, 2001

[23] Christophe Diot, Brian Neil Levine, Byan Lyles, Hassan Kassem, and Doug Balensiefen, *Deployment Issues for the IP Multicast Service and Architecture*, IEEE Network, pp. 78-88, January/February 2000

[24] Chunming Qiao, Myoungki Jeong, Amit Guha, Xijun Zhang, and John Wei, *WDM Multicasting in IP over WDM Networks*, Proc. of 7th International Conference on Networks Protocols, pp. 89-96, 1999

[25] Curt Newton, *The Value of Transport networking in a Data-Centric World*, white paper, Optical Network Group, Lucent Technologies

[26] D. Blokh, and G. Gutin, *An approximation algorithm for combinatorial optimization problem with two parameters*, manuscript, 1995

[27] E. Biersack, and J. Nonnenmacher, *WAVE: A New Multicast Routing Algorithm for Static and Dynamic Multicast Groups*, in proc. 5th Int'l Workshop on Network and Operating System Support for Digital Audio and Video, April 1995

[28] Eytan Modiano, *Random Algorithms for Scheduling Multicast Traffic in WDM Broadcast-and-Select Networks*, IEEE/ACM Transactions on Networking, vol. 7, no. 3, pp. 425-434, June 1999

[29] Ezhan Karasan, and Ender Ayanoglu, *Effects of Wavelength Routing and Selection Algorithms on Wavelength Conversion Gain in WDM Optical Networks*, IEEE/ACM Transactions on Networking, vol. 6, no. 2, pp. 186-196, April 1998

[30] Frank K. Hwang, Dana S. Richards, and Pawel Winter, *The Steiner Tree Problem*, Elsevier Science Publishers, 1992

[31] Fred Bauer and Anujan Varma, *Degree-Constrained Multicasting in Point-to-Point Networks*, in proc. INFOCOM'95, vol. 1, pp. 369-376, 1995

[32] Fred Bauer, and Anujan Varma, *ARIES: A Rearrangeable Inexpensive Edge-Based On-Line Steiner Algorithm*, IEEE Journal on Selected Areas in Communications, vol. 15, no.3, pp. 382-397, April 1997

[33] George N. Rouskas, and Ilia Baldine, *Multicast Routing with End-to-End Delay and Delay Variation Constraints*, in Proc. IEEE INFOCOM'96, vol. 1, pp. 353-360, 1996

[34] George N. Rouskas, and Mostafa H. Ammar, *Multidestination Communication Over Tunable-Receiver Single-Hop WDM Networks*, IEEE Journal on Selected Areas in Communications, vol. 15, no. 3, pp. 501-511, April 1997

[35] G. K. Chang, G. Ellinas, B. Meagher, W. Xin, S. J. Yoo, M. Z. Iqbal, J. Young, H. Dai, Y. J. Chen, C. Lee, X. Yang, A. Chowdhury, and T. F. Chen, *A Proof-of-Concept, Ultra-low Latency Optical Label Switching Testbed Demonstration for Next Generation Internet Networks*, Optical Fiber Communication Conference, 2000, vol. 2, Pages: 56-58

[36] Gokhan Sahin and Murat Azizoglu, *Multicast Routing and Wavelength Assignment in Wide Area Networks*, SPIE Conference on All-Optical Networking: Architecture, Control, and Management, Nov. 1998, SPIE vol. 3531, pp. 196-208

[37] Harry G. Perros, *Queueing Networks with Blocking*, Oxford University Press, 1994

[38] Harry, J. R. Dutton, *Understanding Optical Communications*, Prentice-Hall, 1998

[39] Hwa-Chun Lin and Chun-Hsin Wang, *A Hybrid Multicast Algorithm for Single-Hop WDM Networks*, Proceedings of the IEEE INFOCOM 2001, vol.1, pp. 169-178, 2001

[40] Imrich Chlamtac, Aura Ganz, and Gadi Karmi, *Lightpath Communications: An Approach to High Bandwidth Optical WAN's*, vol. 40, no. 7, pp. 1171-1182, July 1992

[41] Imrich Chlamtac, Aura Ganz, and Gadi Karmi, *Purely Optical Networks for Terabit Communication*, INFOCOM'89, Proceedings of the 8th Annual Joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging. IEEE, vol. 3, pp.887-896, 1989.

[42] ITU-T, *Definition and Test Methods for The Relevant Parameters of Single-Mode Fibres*, ITU-T Recommendation G.650, Oct. 2000

[43] Jack M. Gill, *Lasers: A 40-Year Perspective*, IEEE Journal on Selected Topics in Quantum Electronics, vol. 6, no. 6, pp. 1111-1115, Nov/Dec 2000

[44] James A. McHugh, *Algorithmic Graph Theory*, Prentice-Hall, 1990

[45] James R. Evans, and Edward Minieka, *Optimization Algorithms for Networks and Graphs*, 2nd Ed., pp. 49-54, Marcel Dekker, 1992

[46] Jennifer M. Yates et al., *Wavelength Converters in Dynamically-Reconfigurable WDM Networks*, IEEE Communications Surveys, pp. 2-15, Second Quarter 1999, http://www.comsoc.org/pubs/surveys

[47] Jingyi He, S.-H. Gary Chan, and Danny H.K. Tsang, *Routing and Wavelength Assignment for WDM Multicast Networks*, proc. GLOBECOM 2001, vol. 3, pp. 1536-1540, 2001

[48] Jon Anderson, James S. Manchester, et al., *Protocols and Architectures for IP Optical Networking*, Bell Labs Technical Journal, pp. 105-124, Jan.-Mar. 1999

[49] K. Bharath-Kumar, and J. M. Jaffe, Routing to Multiple Destinations in Computer Networks, IEEE Transactions on Communications, vol. COM-31, no. 3, pp. 343-351, March 1983

[50] Kuo-Chun Lee and Victor O. K. Li, *A Wavelength-Convertible Optical Network,* Journal of Lightwave Technology, vol. 11, no. 5/6, pp. 962-970, May/June 1993

[51] Kurt Mehlhorn, *A Faster Approximation Algorithm for the Steiner Problem in Graphs*, Information Processing Letters, pp. 125-128, vol. 27, 1988

[52] Laxman H. Sahasrabuddhe, and Biswanath Mukherjee, *Light-Trees: Optical Multicasting for Improved performance in Wavelength-Routed Networks*, IEEE Communications Magazine, pp. 67-73, Feb. 1999

[53] M. Imase, and B. M. Waxman, *Dynamic Steiner Tree Problem*, SIAM J. Disc. Math, vol. 4, no. 3, pp. 369-384, Aug. 1991

[54] M. Klinkowski, and M. Marciniak, *Development of IP/WDM Optical Networks*, Proc. LFNM 2001, pp. 84-87, May 2001

[55] M. Kang, *An Optimal Dynamic Multicast Routing Algorithm for Multimedia Applications*, in proc. IEEE Int'l Conference on Multimedia Computing and Systems'97, pp. 79-84, 1997

[56] Masahiko Fujiwara, *Optical Crossconnect/ADM Based Networks*, Lasers and Electro-Optics Society Annual Meeting, vol.2, pp. 111-112, 1994

[57] Mathew Doar and Ian Leslie, *How Bad is Naïve Multicast Routing?*, proc. IEEE INFOCOM, pp. 82-89, 1993

[58] Myoungki Jeong, Chunming Qiao, Yijun Xiong, and Marc Vandenhoute, *Bandwidth-Efficient Dynamic Tree-Shared Multicast in Optical Burst-Switched Networks*, in proc. ICC 2001, pp. 630-636

[59] Myoungki Jeong, Yijun Xiong, Hakki C. Cankaya, Marc Vandenhoute and Chunming Qiao, *Efficient Multicast Schemes for Optical Burst-Switched WDM Networks*, IEEE International Conference on Communications, 2000 (ICC 2000), vol. 3, pp. 1289-1294

[60] N. Katoh, T. Ibaraki, and H. Mine, *An Efficient Algorithm for K Shortest Simple Paths*, vol. 12, pp. 411-427, networks, 1982

[61] N. Sreenath, K. Satheesh, G. Mohan, C. Siva Ram Murthy, *Virtual Source Based Multicast Routing in WDM Optical Networks*, Proc. of IEEE Int. Conf. on Networks (ICON), pp. 385-389, 2000

[62] N. Sreenath, N. Krishna Mohan Reddy, G. Mohan, and C. Siva Ram Murthy, *Virtual Source Based Multicast Routing in WDM Networks with Sparse Light Splitting*, IEEE Workshop on High Performance Switching and Routing, pp. 141-145, 2001

[63] Neil A. Jackman, Sunita H. Patel, et al., *Optical Cross Connects for Optical Networking*, Bell Labs Technical Journal, pp. 262-281, January-March 1999

[64] O. Aboul-Magd, Sandra Ballare, Ewart Tempest, Raj Jain, LiangYu Jia, Bala Rajagopalan, Robert Rennison, Yangguang Xu, and Zhensheng Zhang, *LDP Extensions for Optical User Network Interface (O-UNI) Signaling*, work in progress, July 2001

[65] Paul Bonenfant, Antonio Rodriguez-Moral, and James Manchester, *"IP over WDM": The Missing Link*, white paper, Optical Network Group, Lucent Technologies.

[66] Pawel Winter, *Steiner Problem in Networks: A Survey*, Networks, pp. 129-167, vol. 17, 1987

[67] Q. Sun, and H. Langendorfer, *Efficient Multicast Routing for Delay-Sensitive Applications*, in proc. 2nd Workshop on Protocols Multicast Systems (PROMS' 95), pp. 377-385, Oct. 1995

[68] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, *A Source-based Algorithm for Delay-constrained Minimum-cost Multicasting*, in proc. IEEE INFOCOM'95, pp. 377-385, 1995

[69] R. Ravi, et al., *Approximation Algorithms for Degree-Constrained Minimum-Cost Network-Design Problems*, Algorithmica, pp. 58-78, vol. 31, 2001

[70] R. Widyono, *The Design and Evaluation of Routing Algorithms for Real-time Channels*, Technical Report TR-94-024, UC Berkeley, June 1994

[71] Rajiv Ramaswami, *Optical Fiber Communication: From Transmission to Networking*, IEEE Communications Magazine, 50th Anniversary Commemorative Issue, pp. 138-147, May 2002

[72] Rajiv Ramaswami, and Kumar N. Sivarajan, *Optical Networks: A Practical Perspective*, 2nd Ed., Morgan Kaufmann, 2001

[73] Ryszard Syski, *Introduction to Congestion Theory in Telephone Systems*, in North-Holland Studies in Telecommunication, vol. 4, North-Holland, 1986

[74] S.-P. Hong, H. Lee, and B. H. Park, *An Efficient Multicast Routing Algorithm for Delay-Sensitive Application with Dynamic Membership*, in Proc. IEEE INFOCOM'98, vol. 3, pp. 1433-1440, 1998

[75] S. Raghavan, G. Manimaran, and C. S. R. Murthy, *A Rearrangeable Algorithm for the Construction of Delay-Constrained Dynamic Multicast Trees*, IEEE/ACM Transactions on Networking, vol. 7, no. 4, pp. 514-529, August 1999

[76] S. Subramaniam, A. K. Somani, M. Azizoglu, and R. A. Barry, *A Performance Model for Wavelength Conversion with Non-Poisson Traffic*, vol. 2, pp. 499-506, Proc. IEEE INFOCOM 1997

[77] Stefan Voss, *Problems with Generalized Steiner Problems*, Algorithmica, vol.7, pp. 333-335, 1992

[78] Stefan Voss, *Steiner's problem in graphs: heuristic methods*, Discrete Applied Mathematics, vol. 40, pp. 45-72, 1992

[79] Sung-Jin Chung, Sung-Pil Hong, et al., *Algorithms for the Degree-Constrained Multicast Trees in Packet-Switched Networks*, in Proc. GLOBECOM'98, vol. 2, pp. 1054-1059, 1998

[80] Terji Durhuus, Benny Mikkelsen, Carsten Joergensen, Soeren Lykke Danielsen, and Kristian E. Stubkjaer, *All-Optical Wavelength Conversion by Semiconductor Optical Amplifiers*, Journal of Lightwave Technology, vol. 14, no. 6, pp. 942-954, June 1996

[81] Vachaspathi P. Kompella, J. C. Pasquale, and G. C. Polyzos, *Multicast Routing for Multimedia Communication*, IEEE\ACM Transactions on Networking, vol. 1, no. 3, pp. 286-292, June 1993

[82] Wei S. Hu, and Qing J. Zeng, *Multicasting Optical Cross Connects Employing Splitter-and-Delivery Switch*, IEEE Photonics Technology Letters, vol. 10, no. 7, pp. 970-972, July 1998

[83] Wen-Yu Tseng, and Sy-Yen Kuo, *Hybrid Scheduling for Unicast and Multicast Traffic in Broadcast WDM Networks*, IEICE TRANS. COMM., vol. E83-B, no. 10, pp. 2355-2362, Oct. 2000

[84] Wen-Yu Tseng and Sy-Yen Kuo, *All-Optical Multicasting on Wavelength-Routed WDM Networks with Partial Replication*, Proc. of 15th Int. Conf. on Information Networking, pp. 813-818, 2001

[85] X. Jia, N. Pissinou, and K. Makki, *A Real-time Multicast Routing Algorithm for Multimedia Applications*, Computer Communications, vol. 20, pp. 1098-1106, 1997

[86] X. Jia, *A Distributed Algorithm of Delay-Bounded Multicast Routing for Multimedia Applications in Wide Area Networks*, IEEE/ACM Transactions on Networking, vol. 6, no. 6, pp. 828-837, Dec. 1998

[87] X.-H. Jia, D.-Z. Du, and X.-D. Hu, *Integrated Algorithms for Delay Bounded Multicast Routing and Wavelength Assignment in All Optical Networks*, Computer Communications, vol. 24, pp. 1390-1399, 2001

[88] Xiao-Hua Jia, Ding-Zhu Du, Xiao-Dong Hu, Man-Kei Lee, and Jun Gu, *Optimization of Wavelength Assignment for QoS Multicast in WDM Networks*, IEEE Transactions on Communications, pp. 341-350, Feb. 2001, vol. 49, no. 2

[89] Xijun Zhang, John Wei, and Chunming Qiao, *On Fundamental Issues in IP over WDM Multicast*, Proceedings of 8th IEEE International Conference on Computer Communications and Networks, pp. 84-90, 11-13 Oct. 1999, Boston

[90] Xijun Zhang, John Wei, and Chunming Qiao, *Constrained Multicast Routing in WDM Networks with Sparse Light Splitting*, IEEE INFOCOM 2000, pp. 1781-1790

[91] Xuejun Sun, Shujun Zhang, et al., *DWDM Transmission Systems: Principles and Testing*, China People's Post and Telegraph Publishing House, Feb. 2000 (in Chinese)

[92] Y. M. Lin, W. I. Way, and G. K. Chang, *A Novel Optical Label Swapping Technique Using Erasable Optical Single-Sideband Subcarrier Label*, IEEE Photonics Technology Letters, vol. 12, no. 8, pp. 1088-1090, Aug. 2000

[93] Yuanyuan Yang, Jianchao Wang, and Chunming Qiao, *Nonblocking WDM Multicast Switching Networks*, IEEE Transactions on Parallel and Distributed Systems, vol. 11, no. 12, pp. 1274-1287, Dec. 2000

[94] Zeydy Ortiz, George N. Rouskas, Harry G. Perros, *Scheduling of Multicast Traffic in Tunable-Receiver WDM Networks with Non-Negligible Tuning Latencies*, Proceedings of the ACM SIGCOMM '97 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 301-310, 1997

[95] Isabel Rosseti, Marcus Poggi de Aragão, Celso C. Ribeiro, Eduardo Uchoa, Renato F. Werneck, *New benchmark instances for the Steiner problem in graphs*, in Extended Abstracts of the 4th Metaheuristics International Conference (MIC' 2001), pp. 557-561, 2001