uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Tun Hu
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Dynamic Optical Multicast Routing in Sparse-splitting WDM Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Hussein T. Mouftah
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

A. Jaekel

S. Shirmohammadi

P.X. Liu (absent)

A. Nayak

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Dynamic Optical Multicast Routing in Sparse-splitting WDM Networks

by

Tun Hu

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada

May 2007

Your file Votre référence
ISBN: 978-0-494-32407-3
Our file Notre référence
ISBN: 978-0-494-32407-3

Author's Declaration

I hereby declare that I am the sole author of this thesis.
I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize the University of Ottawa to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Abstract

While most of the previous optical multicast routing and wavelength assignment algorithms for sparse-splitting networks have focused on the static traffic patterns, this thesis concentrated on the algorithm performance in the dynamic traffic environment. A dynamic constrained multicast routing algorithm that is based on a widest-first path heuristics (DCMR-WFPH) has been proposed. On the contrary, all of the other optical multicast algorithms known to date utilize the Shortest-Path-based Heuristics (SPH) during the routing stage. A bottleneck wavelength assignment scheme has also been introduced. Combined with the DCMR-WFPH algorithm, this approach can reduce wavelength search space during the dynamic wavelength assignment stage. Furthermore, in this thesis, a problem in the well-known shortest-widest algorithm has been discovered and presented.

The advantages of the algorithm have been demonstrated by simulation results. Compared to the Member-only algorithm, a well known SPH based tree generation algorithm, our algorithm outperforms in load balancing and blocking probability performances. Experiments have also shown that the method of limiting the number of wavelengths each forest can occupy, termed as forest trimming, can be utilized to improve the blocking probability performance. It has also been shown by simulation experiments that certain sparse-splitting networks equipped with less fanout capacity MC-OXCs can achieve similar performance to networks with more fanout capacities.

Due to the special properties of the widest-first routing algorithm, it is possible that loops can be generated during the tree generation stage. Through formal analysis, in this thesis, a number of properties in the DCMR-WFPH algorithm have been proven. Moreover, it was justified that the DCMR-WFPH algorithm can guarantee to generate real trees: No loops will be produced during the routing process.

Acknowledgements

First and foremost, I thank my thesis supervisor, Dr. Hussein Mouftah for his consistent guidance and encouragement throughout my study. The ultimate support he has been providing, and the confidence he has in me have been driving me from one success to another. I will continuously benefit from his advice throughout my life.

I am also very grateful to the strong and consistent support that my family has given me. Their names are Xingreng Hu, Chaomin Lan, Yareng Hu, Wandi Wu, Rong Hu, and Ronghua Fu. Their helps and understanding have made the study much easier, happier and more meaningful.

I would also like to extend my gratitude to the research group of Dr. Mouftah's in the University of Ottawa. Fellows such as Dr. Pin-Han Ho, Dr. Baoxian Zhang, Dr. Jun Zheng, Richard Paramod, Atousa V. Sichani, Marwin, Shafig, Hongxia Wang, etc. have all been part of the support group for my research.

Many friends have also helped to make my life of graduate study in the University of Ottawa enjoyable as well as fruitful. They are Lily Laiyin Leung, Dr. Binsheng Liu, Chris Yee, Brian Wong, Wenling Wu, Emily Yeh, Clement Chan, Kan Li, Xing Song, just to name a few.

List of acronyms and symbols

AOMH	All Optical Multicast Heuristics
ATM	Asynchronous Transfer Mode
AVC	Active Vertical Coupler
DaC	Drop and Continue
DCG	Dynamic Cost Greedy
DC-MCRWA	Dynamic Constrained Multicast Routing and Wavelength Assignment
DCMR	Dynamic Constrained Multicast Routing
DCMR-WFPH	Dynamic Constrained Multicast Routing with Widest-First-Path Heuristics
EO	Electrooptic
FDB	Full Destination Blocking
FPLC	Fixed-paths Least-Congested
GMCT	Greedy Multicast Tree
GMPLS	Generalized Multiprotocol Label Switching
IP	Internet Protocol
IS-IS	Intermediate System to Intermediate System
LAN	Local Area Network
LCP	Least-congested-path
LMP	Link Management Protocol
LSP	Label-Switched-Path
MAN	Metropolitan Area Network
MC	Multicast Capable
MCG	Multicast Capable Group
MC-OXC	Multicast-Capable Optical Cross Connect
MC-RWA	Multicast Routing and Wavelength Assignment
MC-WC	Multicast Capable with Wavelength Conversion
MI	Multicast Incapable
MIG	Multicast Incapable Group
MI-WC	Multicast Incapable with Wavelength Conversion
MPLS	Multiprotocol Label Switching
MST	Minimum Spanning Tree
NMCF	Nearest MC nodes First
O/E/O	Optical/Electronic/Optical
OBS	Optical Burst Switching
OSPF	Open Shortest Path First
OTMCF	On-tree MC nodes First
OXC	Optical Cross Connect
OXS	Optical Crosspoint Switch
PCU	Path Construction Utility
PDB	Partial Destination Blocking
PSC	Passive Star Coupler
PVS	Primary Virtual Source
QoS	Quality of Services
RWA	Routing and Wavelength Assignment
SaD	Splitter-and-Delivery
SCG	Static Cost Greedy

SE	Switching Element
SOA	Semiconductor Optical Amplifier
SPH	Shortest-Path-based Heuristics
SVS	Secondary Virtual Source
TCM	Tap and Continue Module
TCP	Transport Control Protocol
TDM	Time Division Multiplexing
TIR	Total Internal Reflection
VS	Virtual Source
WA-MNC	Wavelength Assignment to Maximize Network Capacity
WDM	Wavelength Division Multiplexing
WFP	Widest-First-Path
WFPH	Widest-First-Path Heuristics
$G(V,E)$	A directed graph that represents the network
V	The set of the network nodes
E	The set of the network edges
$e(i,j)$	A network edge
W	Number of wavelengths supported by the network
$M(v)$	The maximum fanout capacity of a node v
M	Network maximum fanout capacity
$R(s,D)$	Represents a multicast request
s	Represents the source node
D	The set of destination members
$F(s,D)$	Defines a forest F
$T(s, D')$	Denotes a member tree T of a forest
D'	A subset of the destination members
$m(v)$	The number of children of node v
d_{ij}	Distance of a link $e(i,j)$
b_{ij}	Width of a link $e(i,j)$
B_P	The width of a path P
B_T	Width of a tree T
B_F	Width of a forest F
V_T	Set of nodes allowing more branches
V_T'	Set of nodes that do not allow more branches
D^*	Destination members yet to be included during the algorithm
S	The fraction of MC nodes
T	The fraction of destination member nodes
U_{NET}	Network overall wavelength utilization
U_{Link}	Average link utilization
W_F	Number of wavelengths limit on each forest
G'	Denote a subgraph of G during the algorithm operation

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background

In the past several decades, data communication technology has been experiencing rapid growth, driven by the market demands. The developments can be described from different layers in the TCP/IP protocol architecture. At the application level, various users are requesting more bandwidth, Quality of services, privacy and security. At the Internet layer, the Internet Protocol (IP) has been dominantly utilized such that most of the application data are transmitted in the form of IP packets. As of the year 2000, the volume of IP form data traffic has surpassed voice traffic. Down to the physical layer, optical networking technology has become mature. State-of-the art Optical fibers can operate at up to 50 terabits per second [Gro04], [Yan02]. Between IP and Optical layers, there existed a multitude of switching technologies due to industry evolutions, which includes ATM, SONET, Frame Relay etc. As an effort of simplifying the works of interfacing different switching protocols, Generalized Multiprotocol Label Switching (GMPLS) architecture has been proposed. It can assume a wide range of control functions, including routing, signaling, Quality of Service and traffic engineering [Ban01]. Carrying IP traffic over Optical Networks, with GMPLS providing the control and interfacing services between them, has become the *IP over Optics* model for the Next Generation Internet.

Since current electronic processing technique offers a data rate of a few gigabits per second, it poses a bottle neck for the optical networks that employs *Time Division*

Multiplexing (TDM) scheme. As the counterpart of TDM, *Wavelength Division Multiplexing* (WDM) technology does not require the increase of electronic multiplexing rate. Using the *Optical Cross Connect* (OXC), a device that is able to switch wavelengths purely at optical domain, mesh optical networks can be implemented as *wavelength routed* WDM optical network.

End to end connections in WDM networks are established utilizing wavelength channels. A *lightpath* refers to such an optical channel that may span multiple physical links, without intermediate electronic process [Chl92]. In the absence of *wavelength converters*, each connection can only occupy one single wavelength. This is a condition known as the *wavelength continuity constraint*.

Earlier generation optical networks require *Optical/Electronic/Optical* (O/E/O) conversions at every intermediate node. This kind of networks is referred to as *opaque* networks. The maturing and deployment of the Optical Cross Connect (OXC) technology eliminates the unnecessary intermediate electronic processing. Therefore, end-to-end all optical connections can be established. This kind of networks is called *transparent* network. It has the advantage of preserving data integrity and reducing the electronic processing overhead. However, in a generic optical network, some nodes inevitably are required to undergo O/E/O processes, due to the needs such as *wavelength conversion*, signal regeneration, performance monitoring or other control functions. A *translucent* network refers to an optical network where only a subset of the nodes participates in O/E/O transactions. Researches on how translucent networks achieve optimal performances through limited O/E/O transactions, especially limited wavelength conversions are seen in [Ram97], [Sub98].

At the meantime, Internet applications that require point to multipoint connections, namely, IP multicast applications, are experiencing consistent growth. Many of them, such as HDTV, distributed gaming, medical imaging, and multimedia conferencing etc., are bandwidth-demanding. Optical networks, especially WDM networks, are ideal carriers for these applications. Three scenarios of IP multicast over WDM networks have been previously presented [Zha00]: 1) Multicast purely at IP layer via electronically copying data at every intermediate node; 2) Implement multiple unicast at WDM layer from the source to each destination; 3) Support multicast directly at the optical layer via *light-splitting*. The first option induces unnecessary electronic processing overhead and violates optical data transparency. The second one does not efficiently utilize the wavelength resources. The third scheme is more desirable since it reduces transmission latencies and wavelength usage.

In order to provision multicast directly at the optical domain, an Optical Cross Connect equipped with a passive device like *a power splitter*, can split an incoming optical signal into multiple, up to N outgoing output light channels. The number N is called *maximum fanout capacity*. This kind of OXC is called multicast-capable OXCs (MC-OXCs). More details about MC-OXC architectures will be described in the next Chapter.

The implementation of multicast directly at the optical domain is referred to as *optical multicast*. In the next section, we discuss some of the previous works that have been done in this area, which initially have provided motivations for our work.

1.2 Motivation

Introduced in [Sah99], *lighttree* is the basic logical construct to implement optical multicast. It defines an optical channel stretching from one source to multiple destinations. In the absence of the wavelength converters, it only occupies a single wavelength. In [Zha00], the notion of lighttree has been further generalized to be that of *light-forest* to accommodate the situation (with best effort) where a single lighttree is not adequate to reach all the destinations due to various constraints. A light-forest refers to one or more trees rooted at the same source, which may utilize more than one wavelength. Figure 1.1 shows an example of a light-forest utilizing two wavelengths, stretching from one source to four destinations.

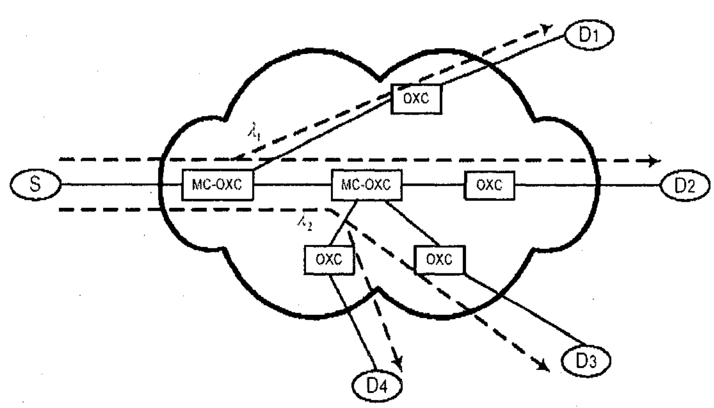


Figure 1.1 A light forest utilizing two wavelengths

To successfully construct lighttrees or light-forests, appropriate routes must be selected and wavelengths must be assigned for all the destinations involved. The essence of this process is the fundamental *routing and wavelength assignment* (RWA) problem in WDM networks, an area that has drawn tremendous amount of research attentions in recent years [Zan00]. The special case of multicast RWA is referred to as *MC-RWA* problems. Similarly as in the point to point RWA case, MC-RWA problems can be classified into static or dynamic scenarios. Static MC-RWA problems are often solved by optimization techniques [Jia01], [Sah98]. As a generalization of the point-to-point RWA case, the optimal solutions for the MC-RWA problems are also NP-complete [Rou03], [Dut00]. Typical researches that addressed real time MC-RWA problems are seen in [Sah98], [Ram02], where the routing and wavelength assignment problems have been decoupled.

An optical network where optical layer multicasting is supported may have all OXCs equipped with splitters, which indicates a *full-splitting* network. Or it may only have some of the OXCs capable of splitting, in which case the network is called a *sparse-splitting* network [Zha00]. The afore-mentioned researches are based on the assumption that the network has full-splitting capabilities, which means any node can be a branching node during the tree construction. The works in [Wu01], [Zha00], and [Rou02] have looked into the MC-RWA problems in sparse-splitting networks, where routing algorithms are *constrained* because trees can only branch on sparsely located MC-OXCs.

The MC-RWA researches known to date on sparse-splitting networks have assumed static traffic patterns with unlimited supply of wavelengths. Little work has been done to address concerns arising from dynamic traffic patterns in such networks. With the rapid growth of the optical multicast applications, a lot of services are required to be set up *on-demand*, e.g., under a dynamic traffic environment. This fact motivated us to define our research work on a different area: the problem of dynamic constrained MC-RWA algorithms. The "constrained" condition in our context means the following: 1) The network is a sparse-splitting one that not every OXC is multicast capable. 2) For each MC-OXC, the fanout capacity is limited. 3) The number of wavelengths supported by the network is also limited, such that congestions will occur in dynamic traffics. These realistic constraints also distinguish our work from other previous works.

1.3 Objective

The main scope of our research is to investigate the problems in the area of dynamic and constrained optical multicast networks, design a new MC-RWA algorithm for the particular type of networks, and evaluate its performance in dynamic traffic environments.

In a dynamic environment, routing and wavelength assignment algorithms are usually evaluated according to their blocking probabilities [Ram98], [Zan01], [Gro04]. It indicates how well the algorithm accommodates dynamic requests on given network conditions. In addition, the algorithm has to make efficient use of limited network resources, e.g., to better utilize existing bandwidths and serve *traffic engineering* purposes, especially in constrained networks. As the application and network technologies evolve, the control functions are further driven to provide more Quality of Services. Current works in the GMPLS protocol stacks embody efforts in all these directions [Ban01].

Our research work aims to meet these performance requirements as much as possible: The primary goal of our research is to create a dynamic constrained MC-RWA algorithm such that it can demonstrate advantages in the areas of our interests: blocking probability performance, network utilization and traffic engineering, as well as providing Quality of Services.

1.4 Contributions

In the area of dynamic constrained WDM multicast, while most of the dynamic algorithms focus on performances in one or two areas among blocking probabilities, traffic engineering and Quality of Services, suffering tradeoffs in other, this thesis proposed an efficient algorithm that demonstrates strengths in all these metrics.

To date, most of the MC-RWA algorithms incorporate the *Shortest-Path-based Heuristics* (SPH) during the multicast routing stage. In sparse-splitting WDM networks, *Member-only* approach [Zha00] is a light tree routing algorithm that utilizes SPH and one that offers a leading performance in wavelength consumption. In contrast, we designed a Dynamic Constrained Multicast Routing algorithm with Widest-First-Path Heuristics (DCMR-WFPH). Under a dynamic traffic environment, the algorithm has been compared with the Member-only approach through simulation experiements. Results obtained from the DCMR-WFPH algorithm showed significant improvements in terms of blocking probability and traffic engineering aspects. Furthermore, DCMR-WFPH has the capacity to offer bandwidth guarantees for dynamic multicast requests in sparse splitting networks.

For the wavelength assignment procedure, a *Bottleneck First-Fit* wavelength assignment scheme has been introduced to co-operate with the dynamic routing algorithm. By minimizing the wavelengths search space, this unique approach makes dynamic wavelength assignment more efficient.

Parameters of the algorithm can be exploited to enhance the blocking probability performance. The method of *trimming forests*— limiting the number of wavelengths each forest can occupy, has been introduced. Experiments have shown that applying the method can improve the performance in terms of blocking probabilities.

One of the unique constraint conditions that DCMR-WFPH considers is that MC-OXCs have limited fanout capacity. Simulation experiements have shown that certain sparse-splitting networks equipped with less fanout capacity MC-OXCs can achieve similar performance to those equipped with more fannout capacities.

Loops are problematic structures in routing sessions, especially in optical domains. We have further examined the operation of the DCMR-WFPH algorithm and formally proved that the routing results from our algorithm can guarantee to be loop-free. Through this means, it has been justified that the DCMR-WFPH algorithm indeed produces trees. On the contrary, little of such formal analysis has been provided in previous related works.

1.5 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 provides a survey about optical multicast. Relevant knowledge and backgrounds that lead to our work are presented. The detailed algorithm is illustrated in Chapter 3. Various simulation experiments and analysis are described in Chapter 4. Chapter 5 provides formal analyses and justifications on algorithm's primary tree generation operation process. Chapter 6 concludes the thesis and outlines potential future works.

Chapter 2 Survey on advances on Optical Multicast Networking

This chapter provides a survey on the latest technology advances related to optical layer multicast. While a variety of topics are addressed, more attentions are paid to issues that are more closely related to this thesis work.

2.1 Introduction

Multicast refers to the technique to transmit information from one single source to multiple destinations. The subject of implementing multicast at IP (Internet Protocol) layer, i.e., IP multicasting, has been widely studied. Protocols such as DVMRP [Pus00], MOSPF [Moy94], and PIM [Car97], have been developed to construct trees in IP layer to support IP multicast applications. As the Internet technology expands, many concurrent multicast applications, such as HDTV, video conferencing, and distributed gaming, have been developed and are bandwidth-intensive. With the advances of Wavelength-division-multiplexing (WDM) technology, optical fibers can now support enormous bandwidth capacity. Supporting IP multicast applications over WDM networks has become a natural choice.

Two approaches can be adopted to implement IP multicast over WDM networks:

1) Build a multicast tree using the existing IP multicast routing protocols, and then map the routing results to the optical layer.

2) Implement multicast directly at the optical layer, taking the optical layer constraints into account. This technique is referred to as *Optical Multicast.*

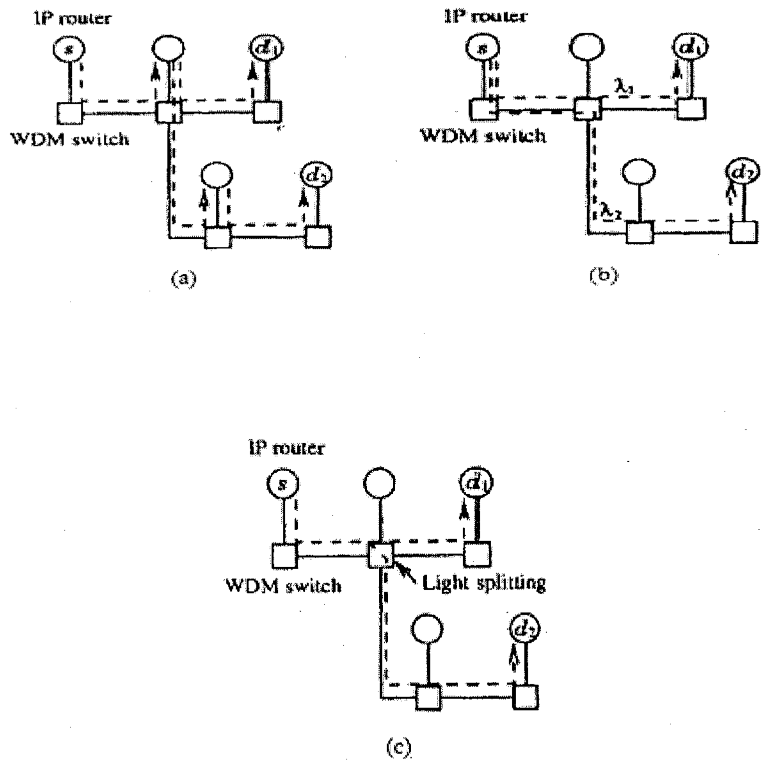


Figure 2.1 IP over WDM multicast

Figure 2.1 illustrates an example that realizes IP multicast over WDM networks using these two approaches. Assume a logical multicast tree has been constructed that spans from the source s to the destinations d_1 and d_2. Figure 2.1 (a) implements multicast purely at the IP layer. Every intermediate router makes electronic copies of the received information and forwards them towards their downstream router, until all destinations are reached. Optical-electronic-optical (OEO) conversions are required at these routers. Thus, this approach has drawbacks in terms of processing latencies and compromising data transparencies. On the other hand, in Figure 2.1 (b), IP multicast is achieved using the *multiple unicast* approach at the WDM layer. The problem with this approach is that when the destination group is large, wavelength channel consumption will become unbearable. In Figure 2.1 (c), multicast is supported directly at the WDM layer: A *light tree* is formed utilizing *light splitting* devices. Since multiple destinations share the bandwidth offered by the same wavelength channel, this approach provides efficient use of wavelength resources.

The first two scenarios in Figure 2.1 are two examples of mapping IP multicast routing results on to the WDM layer. Scenario in Figure 2.1 (c) represents the technique of *optical layer multicast.*

In a unicast session, a *lightpath* is established to connect a single source to a destination at the optical domain. Proposed in [Sah99] and extending the concept of lightpath, a *light tree* defines a clear channel rooted at a single source but reaching multiple destinations. In the absence of the wavelength converters, it only occupies one wavelength throughout the channel. Light tree forms the basic logical construct to accomplish optical layer multicast. When several light trees rooted at the same source, utilizing multiple wavelengths for different trees, they form a more generalized light forest [Zha00].

Optical layer multicast eliminates excessive O/E/O transactions on intermediate routers. Thus it reduces forwarding latencies, preserves data integrities and reduces cost. Furthermore, the routing decisions can be more efficient with a consistent knowledge over the constraints posed by the optical layer.

Optical layer multicast concept can be directly applied to various bandwidth-demanding IP multicast applications such as HDTV, multimedia conferencing, and optical storage area networks (O-SANs). In addition, it can be utilized to offer services to applications such as traffic grooming, 1+1 optical layer protection and provide improved optical network performance [Rou03].

There are two types of WDM networks: broadcast-and-select networks and wavelength-routed multi-hop wide area networks. Both of them can provide optical multicast services.

Broadcast-and-select networks are also referred to as single-hop networks. This type of networks is mostly deployed in Local Area Networks or Metropolitan Area Networks (LAN/MAN). Usually it is composed of a hub-and-spoke topology. Light splitting can be performed via the hub, which is a component based on a passive star coupler (PSC) [Muk97]. A PSC is a passive optical device without any electronic component. It can divide an incoming light signal from any port equally to all the other ports. Stations are connected with each other through the PSC. Each station is equipped with transmitter and receivers, either tunable or fixed. A successful transmission takes place when a transmitter and one or more receivers are tuned to the same wavelength.

Since PSC is a shared media, contentions can occur. For example, when more than one transmitter wants to transmit to the same wavelength channel at the same time, collision is encountered. Another scenario of contention is called *destination conflict*. For a system where stations are equipped with tunable receivers, destination conflict takes place when two or more transmitters intend to transmit to the same destination station on different wavelengths. Therefore, the major challenge for multicasting in a broadcast-and-select network is to design efficient scheduling algorithms to coordinate the tunings. Three categories of scheduling algorithms, namely, random-access based, reservation based and pre-allocation based scheduling algorithms have been proposed. Readers are referred to [Mod99], [Bor95], [Ort97], [Rou97] for detailed discussion on these algorithms.

WDM multi-hop wide-area networks can be further classified into two types: Wavelength Routed and Optical Burst Switching (OBS) networks. Wavelength routed WDM networks are suited for long duration data transmissions and OBS networks are mainly for bursty data traffic. We focus our discussion on wavelength routed multi-hop networks. The term of optical multicast in our work hereinafter refers to the light tree services in multi-hop wavelength routed WDM networks.

By and large, the issues of optical multicast in WDM networks arise from two general categories: issues in data plane and issues in control plane. The issues on the data plane are mainly related to the design of multicast-capable Optical Cross Connects (MC-OXC), The fundamental challenges is to study the hardware architectures that will allow light splitting at lower cost and higher power efficiency [Hu98], [Ali00a], [Ali00b], [Yan00], [Wan06]. On the control plane, the major issue is the multicast routing and wavelength assignment problem [Sah98], [Che02], [He01], [Poo06]. In the following sections, we discuss these issues from these two perspectives.

2.2 Components and challenges on data plane

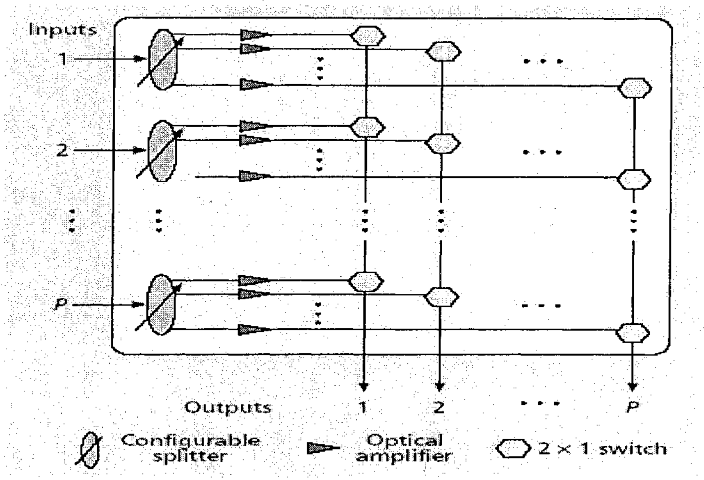
This section addresses the primary issues on data plane of optical multicast networks. The issues are related to the hardware devices that support the light forest services.

2.2.1 Multicast-capable Optical Cross Connects (MC-OXC)

A multi-hop WDM wavelength routed network consists of a number of nodes, i.e., OXCs. They are interconnected by fiber links. In order to support optical layer multicast, light trees must be established among these nodes. The nodes serving as branching points for light trees therefore have to be multicast capable: They are able to replicate the optical signals received from an incoming link to multiple outgoing links. As previously mentioned, these OXCs are called multicast-capable OXCs (MC-OXCs). The key hardware component that enables light splitting in MC-OXCs is the so-called *splitter-and-delivery* (SaD) switch, which consists of *power splitters*.

A power splitter is a passive device capable of splitting optical signals. An n-way power splitter splits an input signal into n identical output signals. The number n is referred to as the *fanout* of the splitter. In an ideal case, the power of each output is $(1/n)$th of that of the incoming signal. Power splitters include *fixed* or *configurable* splitters. A fixed power splitter has a fixed fanout n. On the other hand, a configurable power splitter can split an incoming signal into a variable n_{out} of outgoing signals, $1 \le n_{out} \le n$. In this case, we call n the *maximum fanout capacity* and n_{out} as the actual fanout from the splitter. Compared to the fixed power splitters, configurable power splitters reduce power loss by splitting input signals into the exact number of outputs required by specific applications.

Figure 2.2: A $P \times P$ SaD switch



Power splitters are the key devices in SaD switches. Figure 2.2 shows a $P \times P$ SaD switch proposed in [Hu98], [Ali00a], which utilizes configurable power splitters. It consists of P power splitters, P^2 amplifiers and P^2 2×1 photonic switches. All input ports to the SaD switch carry signals on the same wavelength. Amplifiers are needed to compensate the power loss due to the light splitting. By appropriately configure power splitters and photonic switches, an input optical signal can be split and directed to $1 \leq n_{out} \leq P$ outputs.

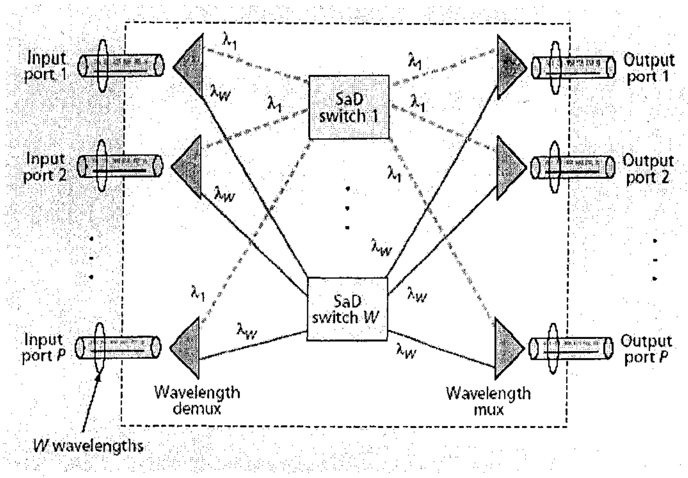


Figure 2.3: A $P \times P$ MC-OXC consisting of W $P \times P$ SaD switches

Figure 2.3 shows a $P \times P$ MC-OXC with W wavelengths consisting of W $P \times P$ SaD switches. Each SaD switch is responsible for the splitting and space switching function for a particular wavelength. Demultiplexers and multiplexers are equipped on each input or output ports to extract or combine different wavelengths. This MC-OXC architecture can establish any permutation of optical multicast or unicast in a strictly nonblocking sense [Ali00a]. However, a large number of power splitters and amplifiers are needed.

The power splitters introduced above result in splitting power losses. Continuous efforts have been made to design multicast capable switches that can counter power losses. Recently, an alternative architecture—Active Vertical Coupler (AVC) based Optical Crosspoint Switch (OXS) Matrix architecture has been proposed. The basic schematic is shown in Figure 2.4. Research has reported that it is able to perform optical multicast without splitting loss. Different from the all optical SaD switch, AVC based OXS is an *electrooptic* (EO) switch. The AVC based OXS matrix consists of two waveguide layers. Interconnected optical switch cells based on AVCs form a switch matrix. At the bottom layer, two passive waveguide are perpendicularly connected in each cell as input and output buses. At the crosspoint two AVCs are formed using a light-amplifying active waveguide layer grown on top of the passive waveguides. A *total internal reflection* (TIR) mirror deflects the optical signal for 90 degrees from the input to the output. Multicast operation is accomplished by turning on more than one switch cells in the same input column of the OXS matrix. Experiments showed that due to the *semiconductor optical amplifier* (SOA) like structure of the AVC active layer, signal is amplified when the AVC operates at supermode. The input signal thus can be directed to multiple outputs at the same or higher power. Thus the light signal splitting is achieved without power loss. Interested readers are referred to [Lee05], [Lee06], [Wan06], [Wan06b], [Wan06c], [Var04] for more details about this architecture.

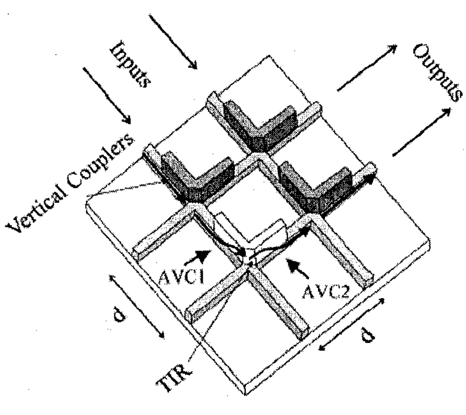


Figure 2.4: Active Vertical Coupler (AVC) based OXS Matrix architecture

2.2.2 Drop-and-continue (DaC) switches

As another switching mechanism that can support optical multicast, *drop-and-continue* (DaC), or *tap-and-continue* (TaC) switch architecture was proposed in [Ali00b]. A DaC switch does not use power splitters. Instead, it utilizes a *Tap and Continue Module* (TCM), shown as in Figure 2.5 to tap an extremely small fraction, e.g., 0.5% of the input signal and direct it to the local station. The remaining signal is guided through a multistage of *switching elements* (SEs) to a desired output. DaC switches therefore do not split light signals among outputs equally and cannot serve as a branching node in a light tree construct. However, it avoids the power loss resulted from splitting operations. Multicast connections with DaC switches can be achieved by determining a path originating at the source, so that the destinations can be sequentially visited along the path. It was shown in [Ali00b] that the task of determining such a path of minimum length is intractable.

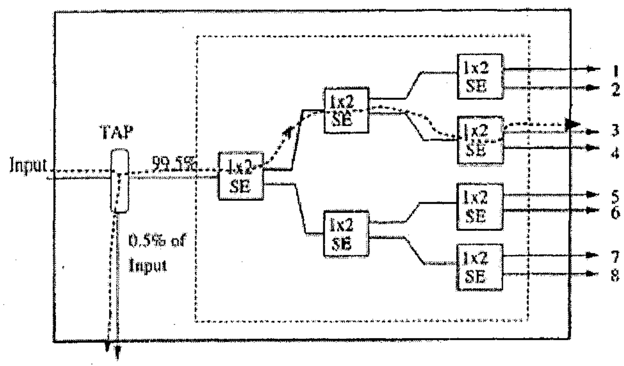


Figure 2.5: A Tap and Continue Module (TCM)

SaD switches can perform splitting up to multiple outputs. Therefore it is a critical component in light tree implementations. However, splitting introduces power losses. The use of amplifers compensates power loss but with increased cost. Furthermore, amplification process also amplifies signal noise. This also applies to the case of AVC based OXS matrix architecture. On the other hand, the DaC switch architecture is cheaper to implement. At the switch, the signal power loss is less than that from SaD switch. The major difference with SaD switch is that only one downstream output link is supported. Considering these two factors, optical multicast can be implemented with a balanced performance and cost using a combination of the SaD and DaC switches.

2.2.3 Design for power efficiency

Regarding the design and deployment of MC-OXCs in a network that supports light tree services, there are two major concerns: power and cost. When optical signals reach destinations, they have to be received at certain power level to ensure quality of service. Inside a light tree, MC-OXCs utilizing SaD switches are the main contributor to the power loss. Therefore, how to design and deploy MC-OXCs to ensure power savings has become one of the primary issues on the data plane. On the other hand, MC-OXCs are more expensive due to their more complicated fabrication compared to OXCs that are not multicast capable. Thus minimizing cost without compromising multicast performance is another major challenge. Both the power and cost issues can be addressed at the level of switch design and network design.

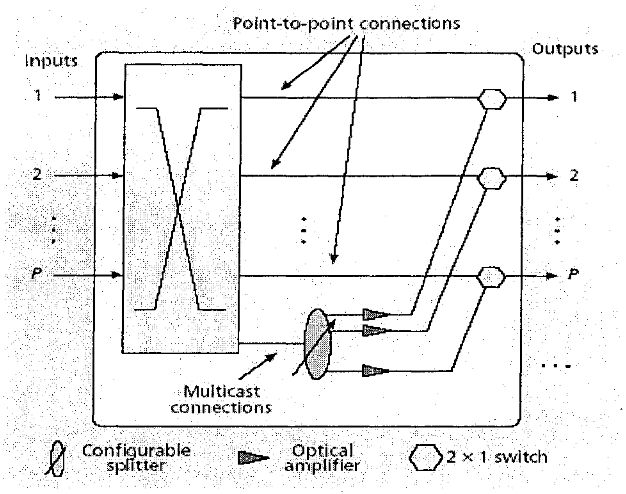


Figure 2.6: A different SaD switch architecture with splitter sharing

At the switch level, a different SaD switch architecture shown as in Figure 2.6 was proposed in [Ali00a]. The design reflects two major differences. First, unicast connections are switched separately and do not traverse any splitters. Secondly, the *splitter sharing* concept was applied. A single configurable splitter is shared by all of the P input ports. Separating unicast traffic from multicast traffic can avoid unnecessary splitting thus avoiding unnecessary power loss. Reducing the number of splitters included in each SaD switch have the advantage of cost savings. However, splitter sharing also reduced the number of multicast sessions the SaD switch can admit. Therefore, the SaD switch design with limited number of splitters are better suited for networks where multicast connections only occupy a smaller fraction of the total number of connections.

On the other hand, at the network level, it has been reported that not all OXCs in the network need to be multicast-capable. The subject of *sparse-splitting* network was introduced and studied [Ali00c], [Rou02], [Zha00]. A sparse splitting network only has a subset of the OXCs that are multicast capable. Reducing the number of MC-OXCs in the network significantly reduces the network engineering cost. However, the concern is how much impact it will cause to the multicast connection requests. Previous study has shown that networks with only about 50% of the OXCs being multicast capable can achieve a reasonably well blocking performance for multicast connections. Another study in this subject was presented in [Yan02]. It was reported that 30% of multicast capable cross-connect is a good trade off between cost and performance in network design. Cost was expressed as the average number of fiber usage in the study.

2.3 Issues on control plane

This section describes some fundamental issues in the control plane of Optical Multicast networks.

2.3.1 General components

In a network where optical multicasts are supported, the primary functions of control plane components are responsible for setting up and managing light-trees (lightpaths). Generally speaking, these functions can be categorized into the following three areas.

1) Topology and resource discovery. Protocols in this area are mainly responsible for the distribution and maintenance of the network state information. The network state information includes optical resource utilization, network connectivity, link capacity and physical layer constraints. Protocol suites such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) have components to maintain these information. In order to support light tree services, network state has to be extended to include

additional information, such as those related to physical layer constraints. For example, the locations of the MC-OXCs in the case of sparse splitting networks, the number of splitters in each node in the case of splitter-sharing MC-OXCs, etc. are constraint information that is crucial for light tree establishment and services.

2) Route computation and wavelength assignment. *Routing and wavelength assignment* problem (RWA) is one of the fundamental control problems in wavelength routed WDM networks. The corresponding problem for the case of optical multicast is referred to as *multicast routing and wavelength assignment* (MC-RWA) problem. In general, two optical layer constraints make the RWA problem more difficult than the traditional IP routing: When there is no wavelength converter in the network, *wavelength continuity constraints* have to be respected, so that a light path must use the same wavelength on all the links along the path; Also, *distinct wavelength constraint* applies to light paths traversing the same fiber links. Optimal solutions to the point-to-point RWA problem have been proved to be NP hard [Dut00], [Zan00]. As a generalization of the RWA problem with multiple destinations to consider, MC-RWA is also NP hard. In addition, MC-RWA has to consider constraints caused by the splitting operation, such as the availabilities and capacities of the MC-OXCs. The network information, namely the *traffic engineering database*, maintained in the topology and resource discovery module provides the condition for the proper operations of the MC-RWA algorithms.

3) Connection management. Protocols in this module deal with the setting up, tearing down of light tree and light path services. Provisioning of protection and restoration issues also belong to this area.

We discuss the subject of MC-RWA, in particular constrained MC-RWA in more details in the following sections.

2.3.2 Multicast Routing and Wavelength Assignment in Full Splitting networks

Similar to the point-to-point RWA case, MC-RWA problems can be solved according to its traffic pattern. In a static traffic pattern, network state is reasonably stable. So a set of light trees can be generated all at once and remain in the network for a long period of time. In dynamic traffic, light tree or light paths provisioning takes place upon the arrival of each request. Concurrent and updated network state information is utilized for each dynamic MC-RWA operation. Usually, there are two approaches to solve the MC-RWA problems: Either decouple the routing and wavelength assignment problems, or consider them together and solve them using some kind of optimization such as linear programming techniques.

In a full splitting network, all OXCs are assumed to be multicast capable (MC-OXCs) and have ample light splitting capacity. Therefore, light trees can branch at any node. Existing IP multicast routing algorithms can be applied at the routing stage. The goal is to build an optimal tree that spans among the source and the destinations. Typically, a cost is assigned to each link of the network. The routing objective is to determine a tree among the multicast group members with minimum total cost. The problem is known as the *Steiner Tree* problem [Hak71], [Hwa92]. The Steiner Tree problem has been proved to be NP-hard when the destination group has more than two members [Gar77]. Therefore heuristic algorithms have been developed. The most well known heuristics are *Shortest Path-based Heuristics* (SPH) [Doa93] and *Minimum Spanning Tree* (MST) based heuristics [Hwa92], [Kou81]. The difference between SPH heuristics and the MST heuristics is that SPH considers the shortest distance between the source and each destination, while MST considers the shortest distance between all pair of members. Note that most of the MST algorithms still utilize SPH to generate routes between node pairs.

After the routing algorithm produces a multicast tree, the wavelength assignment procedure follows. Without the presence of wavelength converters, every path from the source has to be assigned a single wavelength. In this case, a single wavelength can be assigned to all the links of the tree. If no wavelength can be found on all the links of a tree, the multicast request is blocked. Alternatively, multiple wavelengths can be allocated to support a tree. Each wavelength covers a path rooted at the source. Multicast wavelength assignment has been studied as another subject. We will visit this topic again in later section.

In [Lia98], optimization technique was employed to solve the generic Steiner Tree problem. Routing and wavelength assignment are combined into a single problem in the formulation. It was assumed that wavelength converters are available when needed. For a given tree, a cost $C(e,\lambda_i)$ is defined for using λ_i on link e and cost $C_v(\lambda_i,\lambda_j)$ is associated with node v for converting wavelength λ_i to λ_j. The objective was then to minimize the total cost of the multicast tree, including the wavelength-usage cost and the wavelength conversion cost.

Steiner tree problem focuses on building a multicast tree with minimum cost. In addition to considering cost as the objective, optimization techniques have been employed to solve the static MC-RWA problem with other considerations. For example, the work presented in [Jia01] aims to minimize the wavelength usage. A number of heuristic algorithms were proposed. Basically, rerouting and reassigning algorithms were developed to produce trees with less cost and occupying least amount of total wavelengths for the trees. On the other hand, in [Sah98], objective of maximizing the total number of established multicast connections was considered. Bounds on the objective function were presented by relaxing the integer constraints.

In dynamic traffic pattern, multicast request arrives in the network in some random fashion. The network performs MC-RWA in real time and determines if the request can be admitted according to the current network state. Certain admission criteria such as QoS requirements can also be enforced to facilitate the decision making process. If the request cannot be accepted by the network, it is blocked. There are two kinds of blocking policies in this case: *full destination blocking* (FDB) and *partial destination blocking* (PDB). Under FDB, the multicast request is accepted only if all the members in the destination group can be accommodated. The correspondent performance metric for this policy is the *session blocking probability*, which defined the probability that an arriving request is blocked. FDB policy can be applicable for applications such as distributed computing and video conferencing, where all the destinations must be reached for a proper session communication. PDB allows for a multicast request to be admitted even when only a subset of the destination group can be reached. This kind of policy is more suitable for multicast applications such as video on demand services. Performance metric for PDB is defined as the destination blocking probability, which refers to the probability that a destination in the group is blocked.

Since dynamic MC-RWA algorithms have to be simple, the problem is usually solved by decoupling the routing and wavelength assignment procedures. Most of the algorithms still use Minimum Steiner Tree heuristics to build multicast trees at the routing stage. For example, *Fixed MC-RWA*, *Alternate MC-RWA*, and *Dynamic MC-RWA* schemes were proposed in [Sah98]. The basic ideas of these schemes were similar to those discussed in [Zan01] in the case of point-to-point RWA counterpart. First-fit wavelength assignment scheme was applied in [Sah98]. Alternate routing strategy was also employed in [Kam02], [Ram02], where random wavelength assignment scheme was used instead. In these works, each tree occupies a single wavelength after a successful assignment. Full destination blocking policy was adopted.

2.3.3 Multicast Routing and Wavelength Assignment in sparse-splitting networks

Contrary to the full splitting networks, a sparse splitting network only has a subset of its nodes being multicast capable. Nodes in such networks can be grouped into *multicast capable* nodes (MC nodes) and *multicast incapable* (MI) nodes. Optionally, they may be equipped with wavelength converters. In such case, they are called Multicast Capable with Wavelength Conversion (MC-WC) nodes and Multicast Incapable with Wavelength Conversion (MI-WC) nodes.

Since light trees can only branch at MC nodes, MC-RWA algorithms have to consider the constraints caused by the availabilities of these nodes. Extensive research works have been devoted to the area of MC-RWA in sparse splitting networks, e.g., [Zha00], [Wu01], [De06], [Cao06], [Hsi03]. Most of the proposed algorithms assume all nodes, including MC nodes and MI nodes, have Drop-and-Continue capacity. Therefore each node can support at least one downstream child. For each multicast session, the source node is assumed to be able to perform signal replication (splitting) and wavelength conversion at the electronic domain. In other words, a source node can be treated as an MC-WC node. When a single light tree is not sufficient to reach all destinations due to the sparse splitting constraint, multiple trees rooted at the same source, i.e., a light forest, can be utilized to support the multicast session [Zha00].

All these works assumed a static traffic environment in sparse splitting network. MC-RWA problems are decoupled into routing and wavelength assignment procedures. In general, two approaches have been introduced to take the sparse splitting constraints into routing consideration. The first approach builds a multicast forest (tree) using existing multicast routing algorithms such as SPH or MST based algorithms without considering the sparse-splitting constraints. It then reroute the forest according to the knowledge of the MC and MI nodes so that the forest becomes feasible. We call this approach *rerouting* scheme. Alternatively, the routing algorithm can directly utilize the sparse-splitting constraints to construct a feasible light tree from scratch. We call this method as *sparse-splitting constrained routing* scheme. We describe these two schemes in more details next.

2.3.3.1 MC-RWA using rerouting scheme

In rerouting scheme, first, existing multicast routing algorithms are used to produce a multicast forest assuming all nodes are capable of light splitting, i.e., assuming the network is a full splitting one. The routing results are then checked according to the locations of the MC nodes and MI nodes. The initial routing results are infeasible if some MI nodes are serving as branching point in the multicast tree. In this case, all but one downstream path have to be removed from the original tree. Various rerouting algorithms have been introduced to find alternative points for the affected children to rejoin the tree. The most well-known methods are reroute-to-source, reroute-to-any [Zha00], and loop-back [Yan01].

Suppose node v is an MI node. When the original tree shows that v is supporting multiple outgoing links, rerouting is needed for all but one of its children. Let $P(s,v)$ denote the path between the source node s and node v in the original tree. In reroute-to-source, the reverse path of $P(s,v)$ is traced. The first available MC node with enough capacity along that path is selected. The affected children then reconnect to that MC node. Note that in this case at least the source node will be a candidate node to connect to the affected children since it is assumed to be an MC-WC node and have ample splitting capacity, although achieved electronically. In reroute-to-any algorithm, on the other hand, the affected children search across the entire original tree for MC nodes that can support them. The candidate nodes in this case include the source node, any MC nodes, and leaf MI nodes on the tree. When there are multiple paths available for a child waiting to rejoin the tree, the shortest one is chosen. Both rerouting schemes allow multiple trees to be constructed when one tree is not sufficient to reach all the destinations.

The Loop-back algorithm takes a different approach. It assumes all links in the network are bidirectional so that a loop can be formed using the same link with opposite directions. Again an initial tree is infeasible when there are some MI node(s) serving as branching node(s). Suppose node v is such an MI node supporting multiple destinations $d_1(v),\ldots,d_n(v)$. The loop-back algorithm first selects one of the destinations, e.g., $d_1(v)$ to serve. From v, the light signal is forwarded along the path $P(v, d_1(v))$ until it reaches $d_1(v)$. Then it traverses back to node v along the reverse path of $P(v, d_1(v)$ using the links on the opposite direction. Starting from v again, signal travels along another path $P(v, d_2(v))$ to reach the second destination. It then similarly loops back to v using the reverse path of $P(v, d_2(v))$. This process is repeated until all of the destinations that v is supporting are reached. Using the loop-back algorithm, any node with DaC capacity can have multiple outgoing links. Therefore, multicast sessions can be implemented without going through light splitting (MC nodes). However, this scheme leads to longer delay and occupies more light channels. Note that although signals loop back along the same paths, the original multicast routing trees have to be constructed without loops in order for this scheme to work properly.

2.3.3.2 MC-RWA using sparse-splitting constrained routing scheme

Protocols using the sparse-splitting constrained routing scheme construct feasible trees from scratch. Separate rerouting step is thus eliminated. Information about the MC nodes and MI nodes are considered during the multicast routing forest or tree construction stage. A number of such routing algorithms have been proposed in the literature. According to the method they use to generate forest, they can be further classified into the *source-based* and *virtual source (MC nodes) based* multicast routing algorithms. We discuss them in the following.

1) Source-based multicast routing algorithms

This kind of algorithm grows trees starting from the source node of each multicast session. The routing schemes are usually based on some kind of revised SPH or MST algorithms, which take the sparse splitting constraints of MC nodes into account. Examples of the source-based multicast routing algorithms include the Member-First, Member-Only [Zha00] and Greedy Multicast Tree (GMCT) [De06] routing algorithms.

In Member-First routing algorithm, a quasi-Dijkstra's algorithm is utilized to construct the light forest. Different from the original Dijkstra algorithm, candidate links to be included in each step are organized as a priority queue according to the lengths. Higher priorities are given to links leading to destination members when links with equal lengths are present. Each time a link is included, the tree is examined according to the knowledge about MC nodes locations. If an MI node is supporting multiple outgoing links, only the link with the highest priority is retained. The affected downstream links and nodes have to be reconsidered at later steps. When all the members of the multicast session are included, the algorithm stops and prunes the branches leading to non-members.

In Member-Only, multicast tree is grown by including members one at a time, beginning from the member with shortest distance to the source. Thus, there is no need for pruning procedure after all the members are included. Trees are expanded by adding feasible paths one at a time to nodes that still allow new branches. These nodes include the source nodes, MC nodes that still have spare capacity, and the leaf MI nodes. If one tree is not sufficient to reach all the members, additional trees are constructed according to the same procedure.

After the routing algorithms produce multicast trees, wavelengths are assigned to these trees. The concept of *segment* was proposed in [Zha00] to deal with the networks with *no*, *sparse*, and *full* wavelength conversions. A segment is a set of links that have to be assigned the same wavelength. A segment corresponds to a subtree in the case of no wavelength conversions and a link with full wavelength conversions. In the case of sparse wavelength conversions, a segment is obtained by removing intermediate nodes that are not capable of performing wavelength conversions on a path. First-fit wavelength assignment scheme was then applied to the segments. Among Reroute-to-source, Reroute-to-any, Member-First, and Member-Only algorithms, it was pointed out in [Zha00] that Member-Only algorithm requires the least amount of wavelengths. But it has greater delay and computational complexity.

Greedy Multicast Tree (GMCT) routing algorithm also starts tree building from the source node. Similar to the Member-Only algorithm, destination members are considered one at a time. Priorities are given to members that are MC capable. However, the algorithm did not specify how the destinations are ordered.

Trees in GMCT are expanded using a mechanism similar to the MST algorithm, so that a member connects to any node on the tree using the shortest path. The difference is that when multiple candidate paths with equal lengths exist for a member, priority is given to the path connecting to an MC node.

Although GMCT considers the MC nodes locations during the routing stage, it does not prevent MI nodes from having multiple outgoing links. Therefore, it only requires a single tree to span all destination members. The constraints from MI nodes are applied only during the wavelength assignment stage. It is assumed that no wavelength converters are available in the network. When an MI node is supporting multiple destinations in the original tree, additional wavelengths are assigned to paths spanning from the source node to each destination.

2) Virtual Source (MC node) based multicast routing algorithms

A virtual source (VS) node refers to a multicast capable node equipped with wavelength converters (MC-WC node). Virtual source node (MC node) based routing algorithms build trees around these "powerful" nodes. We describe some of these algorithms as follows.

In [Sre00], network nodes are classified into four groups according to their multicast capability and wavelength conversion capacity. They are VS nodes, MC nodes, MI-WC nodes, and MI nodes, respectively. Trees are generated in a manner similar to that of Member-Only. However, when there are multiple shortest paths available for a destination member, preferences are given to nodes with more multicast and wavelength capacities. This protocol encourages the more "powerful" nodes to be better utilized.

Another virtual source rooted multicast routing algorithm was proposed in [Sre01]. The main routing procedure in the algorithm involves two phases: the network partitioning phase and tree generation phase. At the network partitioning phase, the entire network is partitioned into regions based on the location of the VS nodes. Every node in the network is associated with its nearest VS node and establishes a connection to it. Thus, every VS node functions as a regional hub and provides support to nodes in its vicinity. All the VS nodes also pre-establish connections among them using shortest paths. The tree generation phase starts when the multicast request arrives. Upon each request, the source first establishes a connection to its correspondent VS node, which serves as a *primary virtual source* (PVS) node. The PVS then connect to other VS nodes that have one or more destinations in their regions. Those VS nodes are called *secondary virtual source* (SVS) nodes. Destination nodes are then reached from the secondary VS nodes. Since this approach utilizes pre-established tree among VS nodes, the connection set up time when multicast request arrives is shorter. However, it requires resource reservation during the network partitioning phase and thus occupies network resources.

An MC node based tree generation algorithm—All Optical Multicast Heuristic (AOMH) was proposed in [Tse01]. In AOMH, members of a multicast group are divided into multicast capable group (MCG) and multicast incapable group (MIG). If MCG is empty or the fanout constraints are not satisfied, the algorithm selects some proxy MC nodes to be included into the MCG. A multicast tree is constructed first for nodes in MCG using SPH heuristics. MIG is further divided into subgroups according to their distances to the MCG members. Sub-trees are then constructed among MCG members and its associated sub MIG members.

Another similar MC nodes based algorithm was introduced in [Hsi03]. The algorithm also constructs minimum cost tree among MC members. Two heuristics are introduced to connect the MI nodes in the request to the MC nodes tree. In *On-Tree MC node first* (OTMCF) algorithm, the MC nodes tree include all the MC nodes in the multicast request. An MI node in the request connects to the closest MC node on the MC nodes tree. In *Nearest MC nodes First* (NMCF) algorithm, each time an MI node in the request finds a nearest MC node in the network and connects to it. MC nodes tree are constructed and expanded including the MC nodes in the multicast request and the MC node that the MI node selects. However, the scenario where a request does not involve any MC node was not considered in these algorithms.

So far all the multicast routing algorithms in sparse splitting networks have assumed ample fanout capacity for all of the MC nodes in the network. Therefore, the actual fanouts of these MC nodes are only bound by the number of outgoing links of these nodes. However, there are two levels of constraints caused by the MC nodes. At the network level, MC nodes may be sparsely located. At the node level, MC nodes may have limited fanout capacity depending on the hardware implementation. The number of splitters at each MC node may also be limited. Further more, all the works in the sparse splitting networks assumed static traffic settings. In the following chapters, we will provide investigations into the behavior of sparse splitting networks under the dynamic traffic. Limited fanout capacity will also be considered as an additional constraint.

2.3.4 Multicast Wavelength Assignment problem

Research works discussed in previous sections primarily focused on multicast tree routing problem, so that a feasible light forest or tree with less cost can be constructed. Usually heuristic wavelength assignment schemes such as First-Fit are subsequently applied to the routing results. Most of the algorithms use a single wavelength to support a tree or subtree. When there is no common wavelength on all of the links on the tree, the multicast request is blocked. However, it was pointed out [Wan06a] that allowing multiple wavelengths to support a tree can reduce the blocking probability, although the wavelength cost will be increased for individual trees.

Recently, multicast wavelength assignment problem has drawn more research attentions. Given a multicast routing tree, the multicast wavelength assignment problem not only assigns wavelengths, either single or multiple, so that the routing results can be supported by the physical layer, it also aims to achieve some optimal objectives. The problem can be classified according to its static or dynamic traffic pattern assumption. Optimization techniques are often used to solve the static multicast wavelength assignment problem. For example, the objectives in [Poo06] and [Che02] were to minimize the number of wavelength conversions. [Jia01] and [Jia02] addressed the problem of minimizing the number of wavelengths; and the objective in [He01] was to maximize the number of multicast sessions to be admitted. The work presented in [Wan06a] aims to assign wavelengths to links so that maximum number of destinations can be reached, meanwhile trying to minimize the total wavelength cost, as a secondary objective.

Dynamic multicast wavelength assignment problem was studied in [Wan03]. Assuming a dynamic traffic environment, the objective was to assign wavelengths to a multicast session so that the remaining network capacity will be maximized. Therefore, the problem was termed as *wavelength assignment to maximize network capacity* problem (WA-MNC). We discuss the approach in more details in the following.

Define *path capacity* as the number of available wavelengths along a path. *Network capacity* refers to the summation of path capacities for all potential paths in a network. Upon completion of a wavelength assignment for a new multicast request arrival, the network capacity is decreased. Maximizing the remaining network capacity is therefore equivalent to minimizing the network capacity decrease in dynamic wavelength assignments. It was assumed that no wavelength converters are available in the network.

As an online wavelength assignment scheme, the purpose of the WA-MNC was to ensure wavelength assignments can support most future potential requests. Dynamic traffic requests are represented by a traffic matrix **P**. **P** includes a set of unicast and multicast connection requests. A multicast session with M destinations can be regarded as M unicast paths from source to each destination. Therefore, **P** can be assumed as including only unicast path connections so that path capacities can be calculated.

In [Wan03], the WA-MNC problem was proved to be NP-hard. Two greedy algorithms, namely, *static cost greedy* (SCG) algorithm and *dynamic cost greedy* (DCG) algorithm were proposed. The algorithms were based on the concept of *wavelength cost ratio*. In a multicast tree T, assume n_k is the maximum number of destinations that a wavelength λ_k can reach. Define c_k as the network capacity decrease when λ_k is assigned to all the n_k destinations. Given a traffic matrix **P**, c_k can be obtained by counting paths in **P** whose path capacity is decreased by one if λ_k is assigned. The wavelength cost ratio, ϕ_k, for wavelength λ_k is calculated by $\phi_k = c_k / n_k$.

SCG algorithm first computes the wavelength cost ratio for each wavelength. The wavelengths are then sorted in non-decreasing order according to their wavelength cost ratios. Wavelengths are then assigned to the destinations they can reach according to that order. DCG slightly differs from SCG. Each time after a wavelength is assigned, the wavelength cost ration for the remaining wavelengths are recomputed. Then the wavelength with the updated minimum wavelength cost ratio is selected. The process continues until all destinations are reached. DCG has more accurate estimate of the wavelength cost ratio. However, it also introduces higher computational complexity.

Using the concept of wavelength cost ratio, SCG and DCG algorithms select wavelengths such that the impact for the future request is minimized, while reaching maximum number of destinations. However, the network capacity decrease c_k, is calculated from the potential traffic matrix **P**. In a dynamic traffic environment, an accurate potential traffic matrix **P** may be difficult to obtain. Furthermore, in dynamic traffic environment, the path capacity for a path should be unambiguously defined as the number of available wavelengths on the most congested links. This is the approach we take in our research work, which will be introduced in the next chapter.

2.4 Summary

This chapter has provided a survey on related works in optical multicast networks. On the data plane, the SaD and DaC switch architectures have been described. They are the fundamental hardware devices to support multicast in the optical layer. On the control plane, some well known MC-RWA algorithms for sparse-splitting networks have been introduced. They either use the rerouting scheme or the sparse-splitting constrained routing scheme. The latter approach in turn, can be further classified into the source-based multicast algorithms and the virtual source (MC node) based multicast routing algorithms. The latest research activities on the multicast wavelength assignment area have also been presented.

Chapter 3 Dynamic Constrained Multicast Routing and Wavelength Assignment in WDM networks

This chapter studies the problem of Dynamic Constrained Multicast Routing and Wavelength Assignment problem in WDM networks (DC-MCRWA). We first analyze the problem scope and define our objectives. Some problems of the related previous research are investigated. Then the approach to solve the particular problems of interests to achieve the objectives is illustrated. A dynamic constrained light tree routing algorithm—the Dynamic Constrained Multicast Routing algorithm with the Widest-First Path Heuristics (DCMR-WFPH) is introduced. A Bottleneck First-Fit wavelength assignment procedure is further presented.

3.1 Problem scope and objectives

The problem that defines the research topic is Dynamic Constrained Multicast Routing and Wavelength Assignment in WDM networks (DC-MCRWA). The scope of this problem can be analyzed from three dimensions that characterize the problem: 1) the aspect of Multicast Routing and Wavelength Assignment; 2) the "constrained" condition; 3) the "dynamic" environment. Each aspect has associated with it some part of our objectives.

The first dimension of our problem is the Multicast Routing and Wavelength Assignment (MC-RWA) problem. Since MC-RWA algorithms are NP-complete, heuristics are often utilized to solve the problem. The first objective according to this aspect, therefore, is to design a heuristic MC-RWA algorithm. Since light forest is a more general concept, the algorithm aims to construct a light forest.

The presence of wavelength converters is an issue that will further complicate RWA problems, although it can help relax the wavelength continuity constraint. Since wavelength converters are expensive equipment that their excessive use should be avoided, as well we try to focus our attentions on the main aspects of our problem; we concentrate on the WDM networks that do not have wavelength converters. The studies on WDM networks without wavelength converters can provide the fundamental basis for the ones that have wavelength converters.

Another dimension of the DC-MCRWA problem is that the routing and wavelength assignment processes have to make decisions based on the constrained conditions. Constrained routing has recently received a lot of research attentions since these conditions are realistic. In our point of view, the constraints for MC-RWA algorithms are mainly caused by two sources: 1) constraints caused by the network resources conditions, e.g., limited wavelength converters, limited supply of wavelengths, sparsely located MC-OXCs or regenerators etc. 2) constraints that are related to the service requirements, such as signal power requirements, delay requirements or bandwidth requirements.

For the following reasons, we consider the constraint caused by the sparse-splitting networks more practical and desirable. Firstly, splitting naturally introduces power loss so that unnecessary splitting should be avoided as much as possible. When an incoming signal is split m ways, the outgoing signal power is reduced to its $1/m$ in an ideal situation. The addition of amplifiers is not an ideal solution because it also amplifies the noise. Secondly, MC-OXCs are more costly because of its more complicated fabrication. More importantly, previous research study [Ali00b] has shown that sparse-splitting networks can achieve a reasonable performance close to that of the full-splitting ones with only 50% of OXCs being MC-capable. Therefore, the first constraint we consider is that the network is a sparse-splitting WDM network.

For other constraints posed by the network resource conditions, we concentrate on 1) limited fanout capacity for each MC-OXCs; and 2) limited supply of wavelengths. In contrast, most of the previous researches on sparse-splitting networks assume unlimited fanout capacity and wavelengths supply.

Service requirements also create constraints for MC-RWA algorithms. Examples of these requirements are power, delay and bandwidth requirements, which can be regarded as the Quality of Service (QoS) requirements. In the area of MC-RWA algorithms, power and delay requirements have been receiving a lot of research interests [Xin04], [Jia01], [Wu01], [Ham05]. However, little attention has been paid to bandwidth requirement as a QoS service guarantee in MC-RWA research areas. We design an MC-RWA algorithm that will treat bandwidth requirement as a primary QoS constraint. As such, multiple mechanisms can cooperate together to ensure a full list of QoS service guarantees to be enforced.

The dynamic emphasis of our MC-RWA algorithm distinguishes our work from other researches on constrained MC-RWA algorithms. Under this assumption, the algorithm is only invoked after a multicast connection request has arrived. Each existing connection expires after a finite amount of time. The main objectives of the dynamic routing and wavelength assignment algorithms are to efficiently utilize network resources and to maximize the number of connections established [Zha00]. The capacity of how well an RWA algorithm is able to accommodate connection requests is usually indicated by the metric of the blocking probability performance. Most of the research work on dynamic RWA algorithms has been focusing on the blocking probability performance. Without exceptions, the performance of blocking probability is also one of our primary concerns.

We are also interested in another important issue in the dynamic traffic environment: the traffic engineering behavior. The issue of traffic engineering has increasingly gained popularity in optical networks. This can be observed in the development of the MPLS and GMPLS infrastructure [Yet03], [Sai00], [Ban01]. The general goals of traffic engineering include more efficient utilization of network resources, as well as more administrative control over traffic conditions. More specifically, both explicit routing and load balancing problems are classified as traffic engineering issues. In our work, we focus on the objective of balancing the traffic loads across the network. Load balancing not only implies an efficient utilization of network resources such as wavelengths, but also has direct impact on other performance objectives such as blocking probability.

One last but an equally important objective in our design of the MC-RWA algorithm, is to ensure that the routing results do not produce loops. Loop-free requirement is one of the fundamental prerequisite in any tree or forest generation algorithms. We discuss this part in details in Chapter 5.

In summary, the scope of our DC-MCRWA problem combines issues arise from two sub-problems: constrained MC-RWA problem and dynamic RWA problem. Through the implementation of our dynamic constrained MC-RWA algorithm, we hope to achieve a multitude of objectives, including advantageous performances in areas such as blocking probability, bandwidth requirement, load balancing, as well as guaranteeing the loop-free routing results.

3.2 Analytical Approach

Our analysis of the scope of the DC-MCRWA problem indicates that the problem can be regarded as a combination of constrained MC-RWA problem and dynamic RWA problem. Hence, we approach the DC-MCRWA problem from these two angles and explore the possibility of obtaining an integrated solution to achieve our objectives. Since the dynamic feature is unique to our constrained MC-RWA problem, we first provide a more thorough analysis of the dynamic RWA problem.

3.2.1 The problem of Dynamic Routing in WDM networks

As a sub problem of RWA problems in WDM networks, dynamic RWA problems have already been extensively investigated [Zan01], [Gro04]. Two approaches have been most widely adopted: decoupling the routing and wavelength assignment procedures; or treating the network as layered-graphs formed on individual available wavelengths [Zan99], [Gro04], and perform shortest path routing over each plane.

Although we are discussing situations in WDM networks, the routing function can potentially be extended to handle situations in non-optical domains. For example, after performing routing, the results can be utilized by the GMPLS protocols to assign and establish *label-switched-paths* (LSP). This process can be treated similarly as that of the wavelength assignment. Via the GMPLS layer, the algorithm can be utilized to offer services to a variety of interfaces.

On the other hand, little research has shown how the layered plane approach can serve a traffic engineering purpose to balance the network resources, e.g., wavelength utilizations across the network. Decoupling routing process can benefit the traffic engineering purpose since available wavelengths can be considered as an entity of network resources across the whole network. For these reasons, we focus on the first approach where the routing function is decoupled and performed in a separate step before the wavelength assignment procedure. Therefore, we concentrate on the dynamic routing problem in WDM networks as the first separate step.

Three general classes of routing algorithms have been devised and utilized for the dynamic routing environment: *Fixed Routing*, *Fixed Alternate Routing*, and *Adaptive Routing* [Zan01], [Li99], [Low97], [Ram98b]. As *static* routing schemes, the first two algorithms although simpler, do not reflect the network state changes, thus suffering from higher blocking probabilities. Since blocking probability performance is one of our main objectives, we incorporate an adaptive routing algorithm into our design.

Most of the adaptive routing algorithms differ according to the following two criteria: 1) the way in which the network state information is collected and updated; 2) the metric(s) that is used for the optimal computation purpose, which set the criteria for the routes selection.

Network state information can be collected and maintained according to *global*, *neighborhood*, or *local* status [Jue00], [Tan96], and [Zan01]. Each approach has its merits and demerits respectively. We choose a routing algorithm that utilizes global information updates, i.e., a *link state* routing approach, as we shall see later. When link state updates do not take place too frequently, this approach is advantageous in obtaining the optimal results with less complexity. For many dynamic optical multicast applications, such as multimedia conferences, we expect the sessions to remain active in the network for a considerable amount of time span. Furthermore, a routing database with global information will help making traffic engineered routing decision easier. Although network state information maintenance is also vital for a successful connection establishment, detailed discussions in this area fall out of our research scope. During the algorithm route computation, we just need to ensure accurate and updated network state information is available prior to the route selection process. The network state information distribution and update mechanism is assumed to be managed by a separate control protocol.

Routing metrics refer to properties associated with each link status, such as distance, cost, hop count, delay, bandwidth etc. They provide fundamental information for route computation. Different routing metrics have been utilized for the purpose of route selections, as seen in numerous routing algorithms. These routing metrics can generally be categorized as *additive*, *multiplicative*, or *concave* metrics. Intuitively, if the resulting metric of a path is the sum of the metrics from all the constituent links, such as distance or cost, then the metric is called an additive metric. Similarly, it is called a multiplicative metric if the one for a path is the product of those from the links, as in the case of loss probability or reliability. If the metric value for a path takes the minimum of all the constituent links, as in the case of bandwidth, it is called a concave metric.

Different link metrics implies different route selection criteria, which contributes to different objective. Generally speaking, routing algorithms use three approaches in terms of how they utilize routing metrics.

1) Single metric type approach: Only one of the three metric types, namely, additive, multiplicative, or concave is used in the routing process. Most of the *Shortest-Path Heuristics (SPH)* based routing algorithms utilize variants of additive metrics, such as distance, delay, cost, etc. Algorithms that are based on concave metrics have also been proposed and studied, especially in the area of adaptive routing algorithms. For example, *Least-Congested-Path* (LCP) routing [Cha94] and *Fixed-paths Least-Congested* (FPLC) routing [Li99] algorithms both use single concave metric—defined as the number of available wavelengths on the most congested link. Note that the difference between these two algorithms is that LCP makes use of global network state information, whereas FPLC uses neighborhood information.

2) Single mixed metric type approach: Different metric types, usually concave and additive, are mixed into one routing metric using certain mixing functions. For example, in [Ass01], an additive metric—link cost, was defined to reflect the wavelength availability information on each link. Dijkstra algorithm (a SPH heuristic) was then used to determine routes.

3) Multiple metric type approach: Use one type of metric as the primary route selection criteria and another type as secondary, either to help break ties or further optimize the route selection. For example, in [Cha94], SPH is used first and then LCP is

applied for tie breaking. The *Shortest-Widest* algorithm proposed in [Wan96] defines *width*, representing the residual bandwidth from the most congested link, as the primary routing metric. It uses the width information to select *widest* paths and use path distance (delay) as tie breaker for paths with equal width.

All these routing schemes have reportedly shown their merits and demerits. We examine them from our perspective, to determine a solution that will contribute most to our objectives.

Currently, SPH based routing algorithms are still the most popular routing scheme. Numerous literatures have shown its promising performance no matter in dynamic or static routing environments. This approach has its clear advantage in providing "best effort" services for IP connection establishment. Shorter paths can be translated as shorter delay in Quality of Service requirements. In wavelength routed optical networks, shorter distance can also mean occupying less fiber links. However, our objectives include multiple service requirements such as bandwidth, and traffic engineering. Routing schemes have to take extra network state information, such as traffic load or wavelength consumption information into account.

The single mixed type approach such as that presented in [Ass01] suggested a measure to combine wavelength utilization information into the SPH approach. Simulation analysis of the algorithm demonstrated some positive results in blocking probabilities. However, in our point of view, this mixed type approach does not define a deterministic relationship between the optimal route selection criterion and the routing metric. The following Figure 3.1 describes a problematic scenario for their routing algorithm. In the figure, each link is labeled with the number of available wavelengths. In

[Ass01], link cost is defined as the inverse of the number of free wavelengths on the link. Then Dijkstra algorithm is run to help select the least cost, i.e., less utilized paths. Therefore, their routing result will select the lower path, *P(s,i,d)* in the figure. However, *P(s,i,d)* is actually a more congested path compared to the upper path because the link *e(s,i)* causes a bottle neck for that path. This means the algorithm sometimes does not successfully achieve its objective to select a path with more available resources. On the other hand, if the objective of the routing algorithm is a shorter distance, then the lower path should indeed be selected. In this case, though, a simple run of Dijkstra algorithm would suffice.

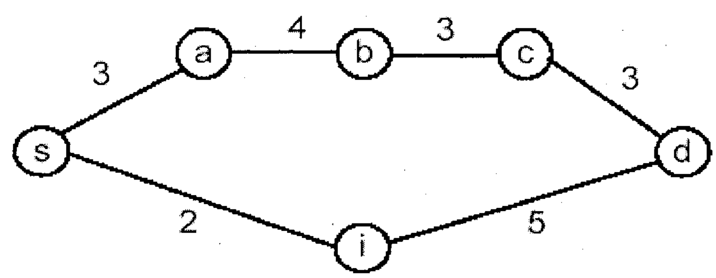


Figure 3.1: A problematic scenario for single mixed type routing algorithms

Since different types of routing metric are associated with different properties of the paths, they should be associated with different routing selection criteria. Mixing different types of metrics into a single entity makes the criteria less deterministic, resulting in less optimal routes. Therefore, the mere benefits of mixing metric types into one routing metric warrant further justifications.

Optical layer connections require end to end wavelength channels to be established. In dynamic traffic conditions, traffic loads are usually distributed non-uniformly. This means congestions often take place on bottleneck links. Once an all optical connection is established, the forwarding delay would not be a primary concern due to the tremendous amount of bandwidth. One of the problems with longer transmission delay is that it leads to power attenuation. This problem has been addressed by numerous previous research work. Their algorithms, as well as other means such as power amplifications, can be combined with our algorithm to achieve a full range of QoS requirements. Therefore, we base our routing algorithm on a concave routing metric.

Some of the proposed routing algorithms that are based on wavelength availabilities, such as LCP and FPLC routings, are implemented in a centralized, or *source routing* fashion. They require a set of feasible candidate paths to be pre-computed. This scenario involves more computation overhead. In our opinion, they do not provide good solutions to our problem settings.

As another routing algorithm that takes residue wavelength resources as the basic routing metric, the shortest-widest algorithm uses width as the primary criterion for route selections and distance as a tie breaker for paths with equal widths. This algorithm intends to provide Quality of Service routings using multiple type metric approach, with bandwidth taking precedence over distance (delay). This approach can better contribute to our routing objectives, i.e., bandwidth requirement, load balancing as well as blocking probability performance. Our implementation of the shortest-widest algorithm adopts the *link state* approach, where the network state information is gathered and maintained globally. The route computation is performed in a distributive manner. We present the detailed description of our implementation later in this chapter.

One of the drawbacks of distributive route computation is that cycles, or loops (used interchangeably), may occur in the results. To cope with this issue, theorem 5 in [Wan96] proved that the shortest-widest algorithm can guarantee loop-free results during the distributive path selection. In our work, we design additional method to prevent loops from happening during multicast tree formations, as we shall see in Chapter 5.

3.2.2 Constrained MC-RWA problems

After analyzing the situation in dynamic routing, we take a close look at the other aspect of the DC-MCRWA problem—the constrained MC-RWA component. A lot of research has been dedicated to the static MC-RWA problem with various constraints assumptions. These results present feasible approaches to be applied to the dynamic routing scenario. The following contexts describe our evaluations on their approaches.

As mentioned earlier, literatures such as those in [Xin04], [Wu01], [Zha00], [Sre00], and [Sre01] represent some of the well-known MC-RWA works in sparse splitting WDM networks. Among them, [Xin04] and [Wu01] regard power quality as their primary objective. Their algorithms involve building some initial tree or forest first and then apply additional method to ensure the destination nodes receive signals with enough and/or fair power quality. Although power quality is another critical issue in the optical multicast area, our focus is on other metrics in the dynamic traffic environment. The proposed algorithm can be used to generate initial trees or forests for their approach. When additional QoS requirements such as power budget are in place, the measures proposed in [Xin04] and [Wu01] can then be employed to ensure the routing results meet the requirements in dynamic settings.

The idea of using *virtual source* or MC node based algorithms to address the sparse splitting networks was proposed in [Sre00], and [Sre01]. Virtual source nodes are essentially more "powerful" nodes that are capable of light splitting and/or wavelength conversions. In these algorithms, the principle ideas were that they encourage the utilization of these powerful nodes, either as branching points [Sre00] or regional hubs [Sre01].

When setting up optical multicast sessions in sparse splitting networks, it is imperative and often advantageous to involve multicast capable nodes (MC-OXCs). The approach in [Sre01] involves resource reservation and pre-setups, among virtual source nodes or inside MC nodes groups. It introduces additional overhead and preoccupies network resources.

Due to power, cost and performance concerns, unnecessary light splitting is discouraged. Also, when multicast capable nodes are inevitably involved in the routing results, we prefer routing results that can avoid excessive splitting, i.e., the results that fan out to less out going channels.

The reasons are further described as follows. Firstly, light splitting contributes to power loss. Although signal power is not our top objective, other conditions being equal, our routing principle selects options that involve less splitting. Secondly, during each session, the source node can conveniently provide the necessary "splitting" or wavelength conversion services. In this case, optical-electronic-optical (O/E/O) conversions take place. While during data forwarding, signal splitting at intermediate nodes can be performed at optical domain to exploit the benefit of all optical connections, electronically processing data at the source is usually mandatory and beneficial. In this sense, the additional overhead caused by copying data electronically and transmit them via available wavelengths should not be regarded as intolerable. Lastly, prioritizing virtual source nodes in the routing algorithm not only render more MC-OXCs to be unnecessarily involved into routing results, it will also inadvertently increase the actual fan out from these nodes. On one hand, more fanout means signal power is further reduced for each outgoing channel. On the other hand, this scenario does not imply better performance in terms of metrics in dynamic settings, e.g., blocking probabilities and load balancing issues. In fact, our study indicates that routing alternatives with less or even no splitting (DaC only) can sometimes achieve similar or equivalent performance in dynamic environment, as we will show in the next chapter.

Four constrained routing algorithms, namely, Reroute-to-source, Reroute-to-any, Member-first and Member-only were introduced in [Zha00]. All these algorithms can be utilized in dynamic routing environment.

To deal with the sparse-splitting constraint in our problem, we incorporate the principle concept in the Member-only algorithm into our algorithm design. The rationale is as follows: 1) Member-only does not involve any pruning, which is simpler to implement in dynamic settings. 2) Member-only does not encourage the use of MC-OXC nodes. It only "discovers" MC-OXC nodes during the route selections. Thus it avoids unnecessary splitting. 3) The research results in [Zha00] suggested that Member-only consumes least amount of wavelengths. Algorithms with less wavelength resource utilization can potentially achieve lower blocking probability performance in dynamic traffic.

In summary, our solution to the DC-MCRWA problem utilizes the fundamentals in the shortest-widest algorithms and the Member-only algorithm. We introduce further mechanisms into our algorithm so that it can meet our objectives in dynamic traffic patterns: better blocking probability and load balancing performance, as well as fulfilling the bandwidth requirement. The algorithm also guarantees to generate trees.

3.3 Problem definition, assumptions and notations

The following list defines the notations in the algorithm and some related assumptions.

$G(V,E)$: Defines a directed graph that represents the network. V is the set of nodes and E is the set of edges or links, interchangeably.

$e(i,j)$: Represents a link, where $e(i,j) \in E$, $i, j \in V$. We assume all the nodes are interconnected by bidirectional links. The existence of a link $e(i,j)$ implies the existence of another link $e(j,i)$ on the opposite direction. We only consider the single fiber case.

W: The number of wavelengths supported on each link, defined as a finite and fixed integer number.

$M(v)$: The maximum fanout capacity of a node $v \in V$. It refers to the maximum number of outgoing channels that a node can split each incoming signal into, or the maximum number of children that the node can support in a light tree.

MC node: Node that is multicast-capable, i.e., node that is capable of splitting and delivering light signals.

MI node: Node that is multicast incapable.

M: Network maximum fanout capacity, a finite and fixed integer, defined as the maximum fanout capacity for all the *MC* nodes across the network. In the same network, we assume all of the *MC* nodes are bound by the same network maximum fanout capacity

M. Therefore, $\forall v \in V$, $M(v)=M>1$ if v is an MC node and $M(v)=1$ if it is an MI node. M represents the limited fanout capacity constraint in our algorithm.

$R(s,D)$: Represents a multicast request, where s indicates the source node and $D \subseteq V$ defines the destination members group in the routing request.

$F(s,D)$: Defines a *forest* F, generated by the routing algorithm according to the request. F is a forest rooted at the source node s and reaching all the destinations in D. It may consist of one or more trees rooted at the same source node.

$T(s, D')$: Denotes a member tree T of a forest F as of the routing result. $T \subseteq F$, and $D' \subseteq D$.

$m(v)$: The number of children of node v. During the routing process, it is always bound by the maximum fanout of the node. Therefore, $\forall v \in V$, $m(v) \leq M(v) = M$ if v is an MC node and $m(v) \leq 1$ if v is an MI node.

d_{ij}: Distance of a link $e(i,j)$, $i, j \in V$. It can be generalized to represent any of the additive metrics such as link cost, or delay.

b_{ij}: Width of a link $e(i,j)$, equivalently denoted as b_e, defined as the number of available wavelengths on the link $e(i,j)$. Similarly it can be generalized to represent other concave metrics such as the residual bandwidth of a link.

B_P: The width of a path. For a directed path $P(i, m) = (i, j, k, \ldots l, m)$, where i denotes the source of the path and m is its destination, $B_P = \min[b_{ij}, b_{jk}, \ldots, b_{lm}]$.

B_T: Width of a tree T, defined as the minimum width of all its constituent paths.

B_F: Width of a forest F, defined as the minimum width of all its constituent trees.

V_T : The set of nodes that is still capable of supporting one or more branches during each tree generation stage.

V_T' : The set of nodes where no more branches are allowed during each tree generation.

D^* : Defines the set of destination members yet to be included during each routing computation.

After defining the notations, we formally state the Dynamic Constrained Multicast Routing and Wavelength Assignment problem.

Definition 3.1: DC-MCRWA problem:

Consider a sparse-splitting network under an IP/GMPLS/WDM infrastructure, where only a subset of network nodes is multicast capable. The nodes are interconnected by bidirectional single fiber links. All of the MC nodes have a maximum fan-out capacity of M. We assume enough splitters are equipped on each MC node, so that it can accommodate multicast connections on multiple wavelength channels. Each link in the network can support up to W wavelengths. The network tries to set up an optical multicast connection upon the arrival of each multicast request R(s,D). If a connection is successfully established, it remains in the network for a finite period of time. The request is blocked if the setup attempt is not successful. The problem of locating the specific channels and assigning wavelengths for each request R(s,D), in order to set up the optical multicast connections that satisfy the specified constraints, is called DC-MCRWA problem. Under certain objectives and constraints, a DC-MCRWA algorithm defines specific methods on how the routing and wavelength assignment are accomplished to solve a DC-MCRWA problem.

The following conditions are further assumed for the algorithm:

No wavelength converters are available in the network. Also "drop and continue" is supported by every node in the network. Thus every node can support at least one downstream child along the same wavelength routed channel where necessary. As a special case, the routing result sometimes appears to be a "path" instead of a conventional tree, where multiple destinations in D are included in the path. They are able to receive the signals through the "drop and continue" process. Note, nonetheless, that a path is also a special tree with only one leaf node.

For every node that happens to become the source node in a request, we assume they can perform "splitting" and wavelength conversions via O/E/O process. That is to say, the source can copy the information and forward it to any number of outgoing links using transmitters. Different outgoing links can use the same wavelength when no wavelength conflicts are encountered. Therefore, the number of children each source can support is only bound by the number of outgoing links that are incident to the source node.

3.4 The DC-MCRWA algorithm

This section presents the details of the Dynamic Constrained Multicast Routing and Wavelength Assignment procedures. Firstly, the Dynamic Constrained Multicast Routing Algorithm with Widest-First-Path Heuristics (DCMR-WFPH) is presented. Some of the unique features about the routing algorithm are discussed. The implementation of the widest-first-path algorithm is described. Some further investigation about the problems on the original shortest-widest algorithm is introduced. It is because of that problem that we call our path generation algorithm widest-first-path-heuristics. We then proceed to propose the Bottleneck wavelength assignment approach.

3.4.1 Dynamic Constrained Multicast Routing Algorithm with Widest-First-Path Heuristics (DCMR-WFPH)

To date, most of the known tree generation algorithms utilize some variants of the Shortest-Path-based Heuristics. Our tree construction algorithm instead is uniquely based on a version of Widest-First-Path Heuristics in order to address issues in the dynamic traffic environment. Throughout the algorithm, we apply the "*width first and distance second*" principle. Destination members are ordered according to their widths and distances to the source. The widest and shortest feasible candidate path is selected for each destination member. Each feasible path is further examined according to the bandwidth requirement. The width information is utilized and updated during the routing procedure to produce load balanced trees.

The algorithm examines the presence of the MC nodes on each selected path similarly as that in the Member-only. The information about the remaining fanout capacity is utilized for further tree expansion. However, we eliminate the destination members that are already included on the path from future consideration.

Upon the arrival of each multicast connection request, the algorithm tries to construct a forest *F(s,D)* between the source node *s* and all the members in the destination group *D*. It takes into account the physical constraints posed by the availabilities and capacities of the MC-OXCs, as well as current traffic engineering database. Forests are formed by adding one tree at a time; and each tree is grown by concatenating paths to nodes that allow new branches, i.e., V_T nodes.

The following Figure 3.2 describes the details of the dynamic constrained multicast routing algorithm with the Widest-First-Path Heuristics approach.

Dynamic Constrained Multicast Routing Algorithm with widest-first-path-heuristics (DCMR-WFPH):

Input: A multicast request *R(s,D)*, which includes the source and the destination membership information; Location and maximum fanout capacity of all the MC nodes; Network topology and current network state information.

Output: If not blocked, return a routed forest.

Step 0: Update the network state information, e.g., the network resource residual capacity information, according to link wavelength availabilities.

Step 1: Begin forest construction. $F(s,D) = \phi$, $D^* = D$. Members in D^* are arranged in descending order according to their widths to the source, where the distances serve as tie breakers.

Step 2: Begin a new tree generation stage. $V_T = \{s\}$, $V'_T = \phi$, and $T = \phi$.

Step 3: Individual path construction (Path Construction Utility). Try to find a widest-first path $P(v, u)$, where $v \in V_T$, $u \in D^*$, which does not involve any node in V'_T and involve only one node $v \in V_T$;

Step 4: If such a path is found:

Step 4.1: If the resulting path $P(v, u)$ has a width B_P of 0 with $v = s$, the session is blocked; otherwise go to step 2 to construct another tree if $B_P = 0$ and $v \neq s$.

Step 4.2: For each valid path, $P(v, u)$ with $B_P > 0$.

Step 4.2.1: Add $P(v, u)$ to the multicast tree T;

Step 4.2.2. Decrease width capacity b_e for every link $e \in P(v, u)$ by one.

Step 4.2.3 For any node y on $P(v, u)$, where $y \neq u$, increase $m(y)$ by 1.

Step 4.2.4: If $v \neq s$, check that if $m(v)=M(v)$, move v from V_T to V'_T;

Step 4.2.5: For any node x on $P(v, u)$, where $x \neq v$ and $x \neq u$, if $m(x)= M(x)$, move x to V'_T; otherwise, move x to V_T;

Step 4.2.6: Move u from D^* to V_T;

Step 4.2.7: Remove from D^* any node that is already included in the path, if $D^* = \phi$, Stop; otherwise, go back to step 3;

Step 5: Else (i.e. Step 3 returns path $P(v, u)$ unfound)

Step 5.1 Include current tree T into $F(s,D)$;

Step 5.2: Go back to step 2 to construct another tree for the rest of the destinations.

Figure 3.2: The Dynamic Constrained Multicast Routing Algorithm with the Widest-First-Path Heuristics approach (DCMR-WFPH)

The algorithm complexity can be expressed by the total number of nodes, N in the network, $N = |V|$. Each multicast request contains at most N destinations. For each destination, the complexity is dominated by the Path Construction Utility. It includes the widest-first path and the sorting algorithm. The complexity for the link state based widest-first path algorithm is equivalent to that of the Dijkstra algorithm [Wan96]. It will be able to generate all candidate paths. The maximum possible number of candidate paths is N. The upper bound for the Dijkstra algorithm is $O(N^2)$. Following that, the PCU then sorts the candidate paths according to the widths. In a general case sorting scenario such as selection sort, it also takes $O(N^2)$ time. Therefore, the worst case time complexity for the DCMR-WFPH algorithm is $O(N^3)$. When better data structure such as priority queue is incorporated in the widest-first path algorithm and the sorting procedures, the time complexity for DCMR-WFPH becomes $O(N^2 LogN)$.

Step 0 of the algorithm in fact can be regarded as another control function that maintains the network state information. The network state information in our algorithm includes the traffic engineering database, e.g., the residue wavelength capacity, as well as the link distance metric on each link. Protocol stacks in the GMPLS suite, such as OSPF-TE and Link Management Protocol (LMP) have already included more traffic engineering database management in their routing function enhancement considerations [Ban01.] In our algorithm, the network state information is maintained according to a link state approach, such that it is distributed and updated globally. DCMR-WFPH algorithm requires accurate and consistent network state information input. Step 0 is therefore in place for that purpose, and it plays a vital role for the algorithm to function properly.

The algorithm also uniquely utilizes the width metric during computations, so that load balanced trees will be produced. In Step 4.2.2, it updates a mirror version of the width information after each path is generated. This process predicts the resources that would be taken by a candidate route. The purpose is to allocate the routes so that the traffic loads across the network will remain as balanced as possible. In case the request is blocked, the load balancing status is unaffected. This mechanism is a unique feature to our algorithm. It prepares the routing results for the wavelength assignment, while trying to balance the overall traffic loads. Note that in DCMR-WFPH, updating width information in Step 4.2.2 only affects the routing results. The actual network state information database remains consistent since a separate control function handles the updates. Again Step 0 in our algorithm updates the global traffic engineering database to ensure the accurate network state information is available for the route computation when the routing procedure begins.

In the DCMR-WFPH algorithm, the sparsely located *MC* nodes have a limited fanout capacity. It is bound by the network maximum fanout capacity *M*. The condition is that $M(v)=M>1$ if a node v is an *MC* node and $M(v)=1$ if it is an *MI* node. After a path is selected, Step 4.2.4 and Step 4.2.5 examine current number of the children each node has. This process ensures the actual fanout from every node does not exceed the maximum fanout capacity constraint. The source node is excluded from the examination. This is consistent with our assumption that the source node can support any number of outgoing links. It performs "splitting" or wavelength conversions as the situation warrants electronically.

Step 4.2.7 removes from D^* the destination members that are already included in a valid path. Since all nodes have the DaC capabilities, these destinations can receive signals through locally dropped signals along the path that traverses them. On the contrary, Member-only algorithm goes over every destination one by one. This process also contributes to the algorithm's loop-free guarantee. We provide the analysis in Chapter 5.

3.4.2 Widest-First-Path Heuristics implementation and the problem of the shortest-widest algorithm

The DCMR-WFPH algorithm utilizes a Widest-First-Path Heuristics (WFPH). With the "width first, distance second" principle, this design is significantly different from other light tree routing algorithms. It significantly contributes to the algorithm's dynamic performance. The Widest-First-Path Heuristics is based on a shortest-widest algorithm proposed in [Wan96].

Some notations that are needed in the WFPH algorithm are described as follows.

Node s: The source node for the algorithm.

Set L: The set of nodes that are so far incorporated in the algorithm. $L \subseteq V$.

$w(i,j)$: Width metric from node i to node j. $w(i,i) = \infty$, $w(i,j) = -\infty$ if node i and j are not directly connected. $w(i,i) = b_{ij} \geq 0$ if there is a link between node i and j.

$d(i,j)$: Distance metric between node i and node j. $d(i,i) = 0$, $d(i,j) = \infty$ if node i and j are not directly connected. $d(i,j) = d_{ij} \geq 0$ if node i and j are directly connected.

$B(n)$: The width of the widest-first-path from the source node s to node n that is currently known to the algorithm. At the termination of the algorithm, this is the width of the widest path in the graph from node s to node n, $P(s,n)$, with its length denoted by $D(s,n)$.

Figure 3.3: The link state based widest-first-path (WFP) algorithm

Widest-First-Path algorithm
Input: source node, topology and current distance and width metric.
Output: The widest-first path to every other node in the graph.

1. $L = \{s\}$, i.e., the set of nodes so far incorporated consists of only the source node. $B(n) = w(s,n)$ for all node $n \neq s$.
2. Determine a set $K \subseteq V$, such that,
   $\{\forall k \mid k \in K,\ k \notin L \ and \ B(k) = \max_{i \notin L} B(i)\}$.
3. If $|K| > 1$, find $k \in K$, such that,
   $D(s,k) + d(k,i) = \min_{j \in K}[D(s,j) + d(j,i)],\ i \in V, i \notin L, and\ j \neq i.$
   Choose k randomly if there are still ties.
4. Include k into L. Also include the edge between k and the node in L that contributes to the maximum width of $B(k)$ to produce the current corresponding widest-first path from s to k, $P(s,k)$. If $|L| = |V|$, the algorithm terminates.
5. For all $i \notin L$,
   If $\min[B(k), w(k,i)] > B(i)$, the widest-first path $P(s,i)$ becomes $P(s,k) + e(k,i)$.
   If $\min[B(k), w(k,i)] = B(i)$, choose the shorter path between the original $P(s,i)$ and $P(s,k) + e(k,i)$ to be the current widest-first path.
   Set $B(i) = \max[B(i), \min[B(k), w(k,i)]]$.
   Go to step 2.

Figure 3.3 describes our implementation of the link state based widest-first-path algorithm. During the update stage in Step 5, WFP algorithm further chooses a shorter path when a tie is encountered. This is a major improvement from the original shortest-widest algorithm.

As a multiple metric types routing algorithm, the well-known distributive shortest-widest algorithm proposed in [Wan96] intends to compute the widest path between a given source and destination pairs. The distance metric serves as a tie breaker. Although it was never formally stated, the "shortest-widest" algorithm implies that it can always achieve its objective: to choose the widest path between a pair of nodes, and the shortest one among the widest ties.

The algorithm works in most of the situations. However, it was never formally proven that it can guarantee to achieve that objective. In fact, our research into this algorithm indicates that the algorithm sometimes cannot meet the "shortest-widest" objective.

Thanks to its link-state routing approach, the algorithm can produce the widest path between a pair of nodes. However, under certain circumstances it cannot guarantee that the shortest path among the widest ties can be successfully selected. This is the reason we call our path generation algorithm as "widest-first-path" algorithm. We demonstrate our observation by creating an example scenario.

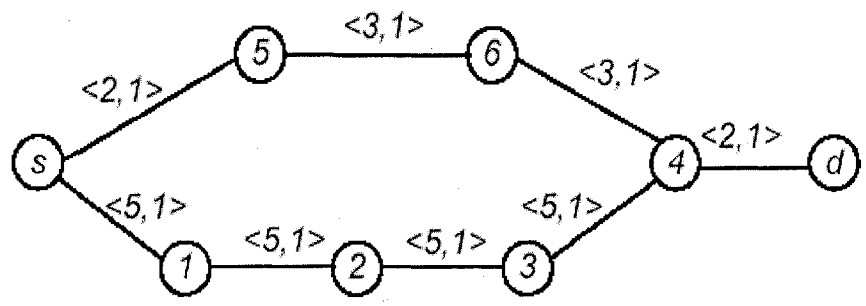


Figure 3.4: A problematic graph for the shortest-widest algorithm

Figure 3.4 depicts an example that presents a problematic scenario for the shortest-widest path algorithm. In the figure, every link is bidirectional. And each link is labeled by the pair of its width and distance metric as <*width, distance*>. For example, the link *e(s,1)* is labeled by <*5, 1*>. That means the link *e(s,1)* has a width of 5 and the distance of the link is 1.

Under the shortest-widest path algorithm, the shortest-widest path for the pair of nodes (*s, 1*) is given by the path of *P(s, 1)* with a width of 5. We denote the result by another 2-tuple, as <*5, (s, 1)*>, where 5 is the width and *(s, 1)* describes the corresponding shortest-widest path. To keep the illustration simple, we omit the resulting length of the path in the notation since it is simply the hop counts of the paths. Similarly, the result for *(s, 2)* is <*5, (s, 1, 2)*>; for *(s, 3)* is <*5, (s, 1, 2, 3)*>; for *(s, 4)* is <*5, (s, 1, 2, 3, 4)*>; for *(s,* *5)* is <*3, (s, 1, 2, 3, 4, 6, 5)*>; for *(s, 6)* is <*3, (s, 1, 2, 3, 4, 6)*>; and particularly for *(s, d)* is <*2, (s, 1, 2, 3, 4, d)*>.

The problem lies in the result for the source and destination pair of *(s, d)*: The shortest-widest path algorithm produced the path of *(s, 1, 2, 3, 4, d)* with a width of 2. However, by a simple inspection of the figure, we can identify that another path, *(s, 5, 6, 4, d)*, will be able to give a path of the same width of 2 and yet with a shorter length.

The algorithm performed properly as how it was designed and intended. The problem arises from the particular structure of the graph and the network state information it currently possesses. It captures the weakness of the algorithm that the distance metric is only in effect during the tie breaking process. The width of the path *(s, 1, 2, 3, 4)* dominates the algorithm operation so that the result of every node pair is determined by the path. The path of (*s, 5, 6, 4*) was never qualified to enter a tie breaking or updating stage in order to affect the result for the node pair of (*s, d*).

The counterexample proved that there are circumstances where the distributive shortest-widest path algorithm fails to produce a widest AND shortest path between a given pair of source and destination nodes. Therefore, we more precisely refer to our implementation of this algorithm a *widest-first-path-heuristics*. Although this problem could be solved by using the source routing approach to examine all possible candidate paths, the need and benefits for the distributive routing algorithm still exist. Therefore, the question remains that whether or not an effective distributive shortest-widest path algorithm exists. Our conjecture is that the problem of using a single distributive routing algorithm to determine a widest and at the same time shortest path between a pair of nodes is an NP-complete problem. More investigations on this point are left to future research.

Although the widest-first-path algorithm may not be able to produce paths that can be both widest and shortest in certain cases, it effectively guarantees that the result is at least a widest one. Therefore, WFP algorithm still contributes significantly in our DCMR-WFPH algorithm so that our objectives can be achieved.

3.4.3 The Path Construction Utility

The Widest-First-Path Heuristics is utilized in one of the main procedures in our algorithm: the Path Construction Utility (PCU). It is responsible for each route selection. PCU also fully embodies our "width first, distance second" design principle. It also handles the remaining fanout capacities of the nodes it encounters. Figure 3.5 provides the full description of the PCU.

PCU in our algorithm offers a number of unique features to our algorithm. It not only excludes the nodes from V_T', but also allows only one node from V_T to be included into the new path. Since current width is the primary routing metric, it is possible for the same link or path to be included into a tree repeatedly. By allowing only one node from V_T to be involved into the new path, the algorithm ensures the routing results do not take duplicate paths or links into the same tree. It also guarantees the routing results are real "trees": No *loops*, or *cycles* (used interchangeably), will be produced in the process. Step C.1.2 in the PCU further contribute to the routing result by returning a shorter and potentially wider path. We analyze these properties in greater details and provide formal justifications in Chapter 5.

Path Construction Utility (PCU):
Step A: Obtain the widest-first-path for every node pair of *(v, u)*, where $v \in V_T$, and *u* is the current destination member to be included into the tree.
Step B: Order the resulting candidate paths according to the widths to *v*, where the distances serve as tie breakers.
Step C: For each of such path *P(v, u)*;
Step C.1: Trace each node *x* on the path in the reverse direction of *P(v, u)*.
Step C.1.1: If any of the node *x* on *P(v, u)* is determined that $x \in V_T'$, the path is invalidated and go back to step C to examine another candidate path.
Step C.1.2: If $x \notin V_T'$, and $x \in V_T$, return *P(x, u)* as the selected path and the procedure stops.
Step C.1.3: Return path-unfound if all candidate paths are exhausted and invalidated.

Figure 3.5: The details of the Path Construction Utility (PCU).

3.4.4 QoS bandwidth guarantee feature about the algorithm

Another important feature of the DCMR-WFPH is that it can provide a bandwidth guarantee for admission control purpose. For many multimedia multicast applications, bandwidth requirement is one of the primary Quality of Service requirements. Therefore the DCMR-WFPH includes a mechanism that checks the routing results according to the bandwidth requirement. This is accomplished in Step 4.1 in the algorithm. To formally justify the mechanism, we first introduce the following assertion.

Assertion 3.1: The resulting forest of the DCMR-WFPH algorithm has a width of at least B if and only if every of its constituent path in each of its member trees has a width of at least B.

Proof: First prove that the forest has a width of at least B only if every of its paths have a width of at least B. The width of a given forest is given by the tree with the minimum width B_T, which is in turn given by the path with minimum width B_P, as defined. Suppose that there exists a forest with a width of at least B, and there exists some path P in the forest with its width less than B, that is, $B_P < B$. Then according to the definition, the width of the tree T consisting of P, B_T would be governed by $B_T = B_P < B$, and then it concludes that the forest width would be bound by the minimum width given by the tree T with $B_T = B_P < B$, which contradicts the assumption.

Following the definition of the width of a forest, it is trivially true that if every path in the forest has a width of at least B, then the forest has a width of at least B.

This completes the proof of Assertion 3.1. ■

Step 4.1 of the algorithm checks the minimum width requirement during each path concatenation. These paths are the basic units that compose a member tree and then forests. By ensuring each path satisfies the minimum width in Step 4.1, the algorithm guarantees the bandwidth for the multicast requests. Assertion 3.1 provides such justification.

In the algorithm, bandwidth requirement is treated as the number of wavelength channels. By modifying the definition of width metric, DCMR-WFPH algorithm can be easily extended to handle bandwidth requirements or any other concave metric in the generic sense. When a multicast request arrives with a QoS criterion demanding a minimum bandwidth of *B*, DCMR-WFPH algorithm can set the minimum width requirement to be *B* instead of *0*.

Note that Step 4.1 also distinguishes the cases where the origin of the path is the source of the multicast request from other cases, so that a deadlock scenario in the algorithm can be prevented.

The following assertion is an extension of Assertion 3.1.

Assertion 3.2: The forest has a width of at least B if and only if every of its links has a width of at least B.

Proof: According to Theorem 4 proved in [Wan96], a path has a width of no less than *B* if and only if each of its links has a width of no less than *B*. While our definition of link width is a special case of residual available bandwidth, combining Theorem 4 from [Wan96] and our Assertion 3.1, it follows that a forest has a width of at least *B* if and only if all its links have a width of at least *B*. ■

Assertion 3.2 complements Assertion 3.1. It establishes the relationship between the basic routing metric and the bandwidth guarantee that the routing results from the DCMR-WFPH algorithm can offer.

3.5 Bottleneck First-Fit Wavelength Assignment

After the DCMR-WFPH admits a request and produces a forest successfully, the next step is the wavelength assignment procedure. Since we assume no wavelength converter is available, all of the links of a particular tree can only occupy the same wavelength.

Wavelengths are assigned to trees according to the order in which they were generated. The routing and wavelength assignment session is blocked if any wavelength assignment of these trees is unsuccessful. In other words, the blocking policy in our DC-MCRWA algorithm is the *full destination blocking* (FDB) scheme, where a multicast session is only admitted when all the destinations members can receive service. It is a more strict sense of blocking policy. On the other hand, the so called partial destination blocking (PDB) allows a multicast session to be admitted even when part of the destination members can be served.

When trees in the same forest have one or more links in common, they have to be assigned different wavelengths to avoid wavelength conflicts. Nonetheless, we allow different trees to utilize the same wavelength if they are link-disjoint. The source node copies the data electronically to accommodate the multicast session using multiple transmitters.

As another unique feature of our wavelength assignment procedure, we introduce a *Bottleneck wavelength assignment* approach. With this approach, we only need to search for available wavelengths from a *bottleneck link* on a tree.

For a path, $P(i, m) = (i, j, k, \ldots l, m)$, a *bottleneck link* for the path is a link e on the path, where $b_e = \min[b_{ij}, b_{jk}, \ldots, b_{lm}] = B_P$. That is to say, the width of a bottleneck link defines the width of the path. During the DCMR-WFPH routing computation, the bottleneck link attribute is recorded after a valid widest path is generated. When there are multiple links that have the equal minimum width for a path, any one of them can be selected to represent the bottleneck link for the path.

DCMR-WFPH algorithm constructs trees by concatenating paths. Therefore, a bottleneck link for a tree is a link that has the minimum width attribute among all the bottleneck links of the constituent paths of the tree. Similarly, a bottleneck link of a tree is obtained as one of its properties after a tree is constructed. Again when multiple bottleneck links exist for a tree, any one of them can represent the bottleneck attribute of the tree. The width of the bottleneck link for a tree also defines the width of the tree.

When the Bottleneck wavelength assignment procedure starts, the algorithm only examines the wavelengths that are available on the bottleneck link. On the contrary, other wavelength assignment algorithms search the entire wavelength spectrum that the network can support, which is much less efficient. In fact, the number of wavelengths that a bottleneck link can support defines the upper bound for the wavelength assignment search space at the time when the procedure starts. When the network does not have wavelength converters, the *wavelength continuity* constraint requires that a wavelength has to be available on all the links for it to become a candidate for the assignment. In the dynamic environment, a successful wavelength assignment inevitably involves a common wavelength that can be found on the bottleneck link.

The above reasoning leads us to the following proposition:

Proposition 3.1: *In the DC-MCRWA algorithm, the width of the bottleneck link for a given tree defines the upper bound for the corresponding wavelength assignment search space. It also defines the maximum number of wavelength channels that a routed tree can support.*

Among all the candidate wavelengths on the bottleneck link, we apply the *First-Fit* wavelength assignment scheme to select the first available wavelength that can span across the entire tree. Therefore, our wavelength assignment approach is termed *Bottleneck First-Fit* wavelength assignment. Note that in dynamic wavelength assignment procedures, the Bottleneck wavelength assignment approach can be combined with any other wavelength assignment scheme such as *Random*.

In a dynamic traffic environment, the traffic loads are unlikely to be uniformly distributed across the network. The Bottleneck wavelength assignment approach effectively reduces the search space to the level as bound by the bottleneck links, avoiding unnecessary check of the entire wavelengths database. By taking the advantages offered by the DCMR-WFPH algorithm and the traffic engineering database, the Bottleneck wavelength assignment scheme offers a unique approach pertaining to the dynamic traffic pattern.

3.6 Summary

This chapter has presented the solution to the DC-MCRWA problem. The approach has considered the aspects of the dynamic RWA problem and the constrained MC-RWA problem. During the analysis on the dynamic routing algorithms, a problem with the single mixed type routing approach is described.

With the "width first, distance second" principle, the DCMR-WFPH, as well as the underlying widest-path-first algorithms have been presented in full detail. Furthermore, as an important discovery, it has been proven that the shortest-widest algorithm cannot always guarantee to produce a widest and at the same time shortest path between a given pair of nodes. We have also justified that the DCMR-WFPH algorithm can be utilized to fulfill the bandwidth requirements for multicast requests. For the wavelength assignment procedure, the bottleneck wavelength assignment approach has been proposed. This novel approach utilizes the results offered by the DCMR-WFPH routing algorithm so that the search space for the wavelength assignment procedure can be reduced.

Chapter 4 Simulation Study and Analysis

Following the introduction of the algorithm, its dynamic performance is evaluated through simulation. In this chapter, simulation results and analysis are presented. Three areas of the algorithm behaviors have been studied. First, the proposed algorithm is compared with the Member-only under a dynamic traffic environment regarding blocking probability and load balancing performance. A method to "trim the forests" is then introduced and the overall blocking probability behaviors observed. Finally, simulation experiments are used to investigate the effects caused by the limited maximum fanout capacity constraints in the sparse-splitting networks.

4.1 Simulation settings and performance metrics

The algorithm is simulated on a fixed NSFNET topology with 14 nodes and 22 links. A simulation tool has been developed to evaluate the performance of the algorithm. The network topology is shown as in Figure 4.1. All the link distance between each pair of the nodes is assumed to be 1. As described in the last chapter, nodes are interconnected by bidirectional fiber links. A fixed and limited number of wavelengths are supported on each link. We only consider the single fiber case.

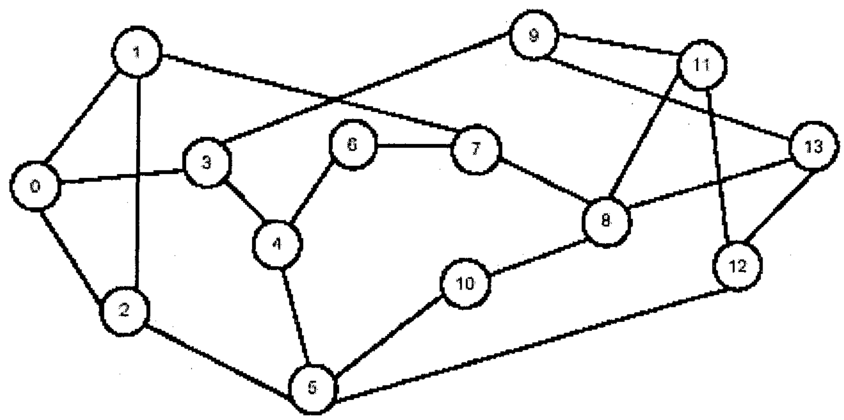


Figure 4.1: The 14 nodes NSFnet network topology.

Four environment parameters are defined as follows.

W: The number of wavelengths supported on each link, defined as in Chapter 3.

M: The network maximum fan-out capacity for all the *MC* nodes, defined as in Chapter 3.

S: The fraction of the nodes possessing splitting ability (*MC* nodes), $0<S<1$.

T: The fraction of nodes to become destinations for each multicast request, $0<T<1$.

While the simulations are run under various loads, these four parameters are fixed during each set of experiments. The sparsely located *MC* nodes are randomly determined according to *S*. After that, we assume they are fixed. Each *MC* node is equipped with enough number of splitters to accommodate all multicast sessions on the incoming links. The multicast requests arrive at the network following a Poisson distribution with the mean inter arrival rate being λ. The network handles one request at a time using the specified MC-RWA algorithm. For each admitted multicast session, the holding time follows an exponential distribution with a mean service rate of μ. Thus the traffic load is given by $\rho=\lambda/\mu$. Different load levels are obtained by keeping λ constant and varying μ.

Upon the arrival of each request, the source and destinations are randomly determined with a uniform distribution. Each multicast session contains a single source and one or more destinations. The nodes to become destinations follow the probability of *T*. When the network resources permit, a node can participate in several multicast sessions simultaneously.

In order to compare the performances of our algorithm with the Member-only, we evaluate two metrics in the simulation: the blocking probability and the network load balancing performance. The statistics of the overall network utilizations are also measured for reference. We are mainly interested in the blocking probability performance in the other two areas of simulation study.

Simulations are run long enough until the statistics of these metrics converge. For example, a total of 2×10^6 requests were generated with performance metrics being measured in batches of 2×10^5 requests. These batches are long enough so that their results are not correlated. In each batch, the blocking probability measurement is given by the number of blocked requests divided by the number of requests after the network behavior is converged under a specific load. The mean value over a succession of 5 independent runs is retained for each data point. 90% confidence intervals are calculated for the data points using the method described in the Appendix A. These confidence intervals have been found very small; not significantly visible on the different curves. As such, they have not been put on all presented plots.

To evaluate the load balancing performance across the network, we first obtain the network overall wavelength utilization and then estimate the percentage of links that are within the range of +/-10% of the overall utilization, after the network behavior is converged. The network overall wavelength utilization is given by the following formula:

$$U_{NET} = \frac{\sum_{l=1}^{L}\sum_{\lambda=1}^{W} O_{\lambda}^{l}}{W \times L} \qquad (1)$$

Where $O_{\lambda}^{l}=1$ when a particular wavelength λ on a link l is being used and 0 otherwise; W is as previously defined and L is the total number of links. Approximate U_{NET} is given by averaging a number of measurements after the network status converged under a specific network and traffic load setting.

For each link, its utilization is given by:

$$U_{LINK} = \frac{\sum_{\lambda=1}^{W} O_{\lambda}^{l}}{W} \qquad (2)$$

We calculate the percentage of links that are within +/-10% of the U_{NET} after the network traffic behavior converged, and obtain the average over a number of such measurements as the estimate.

Note that in order to establish light paths and light trees, a resource reservation process is needed. In general, resources can be reserved on each link in *parallel*, or on a *hop-by-hop* basis along the channels. The hop-by-hop reservation can either be accomplished on a *forward* or a *backward* manner. Light tree signaling and reservation is another research area that is left for further research. It is beyond our research scopes in this thesis. In order to concentrate on the objectives of our research, we assume the parallel reservation scheme is utilized to set up light trees. Based on the link-state routing protocol, parallel reservation scheme assumes each node maintains global network state information. Thus this assumption corresponds to our problem settings. Light tree set up times are therefore exempted from performance evaluation in our research. As such, we are not concerned with resource competitions from different multicast sessions. In other words, we only consider blockings occur during the routing and wavelength assignment process.

4.2 Compare the algorithm with Member-only

Currently, most of the multicast routing algorithms still incorporate the Shortest-Path Heuristics (SPH) in their tree construction operation. One of the main differences between our algorithm and other approaches is that our tree generation process is based on the Widest-First-Path Heuristics (WFPH). The Member-only algorithm is one of the well known constrained multicast tree routing algorithm that utilizes SPH. It offers a leading performance on wavelength consumption. In order to investigate the advantages offered by our algorithm, we compare the simulation results with those from the Member-only in a dynamic environment.

Two groups of typical simulation data are presented in Figures 4.2-4.7. In these sets of experiments, the fraction of the *MC* nodes were set to be $S = 30\%$ and $S = 50\%$ respectively. The settings chosen correspond to the literature study that the networks with 30% and 50% of nodes being multicast capable offer a reasonable tradeoff in terms of cost and performances. The other parameters were set at $M=3$, $T=50\%$, $W=16$. Note that when M is set to be 3, all the *MC* nodes have ample splitting capacity since the network only has a maximum nodal degree of 4.

Figure 4.2 shows the resulting blocking probability for both DCMR-WFPH and Member-only under various loads when S = *30%*. Each curve indicates non-blocking until a sharp increase occurs. We define the first load level where the blocking probabilities significantly rise as *saturation load.* The reason for this behavior is that both algorithms try to accommodate all of the multicast requests at the expense of multiple wavelengths using light forests. Therefore the wavelength resources will be exhausted when the load reaches a certain level. While Member-only behaves only slightly better at loads higher than saturation load (9.0), DCMR-WFPH clearly outperforms Member-only before that: It accommodate all requests up to the load of 8.4. Meanwhile Member-only can only guarantee setups until the load of 6.6. The argument still holds for the case when S = *50%* as shown in Figure 4.3. At 10.2, the saturation load for the DCMR-WFPH is still higher than that from Member-only, which is at 8.4. Under the same algorithm, the saturation loads for S = *50%* is higher than that for S = *30%*. That is reasonably to say, when the network has more *MC* nodes available and all the *MC* nodes have ample splitting capacities, it will be able to accept more active multicast requests in dynamic traffic environment.

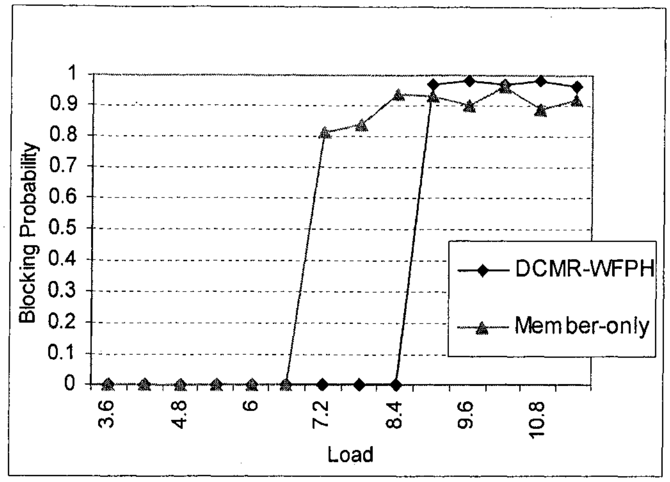


Figure 4.2: Blocking Probability When S=30%.

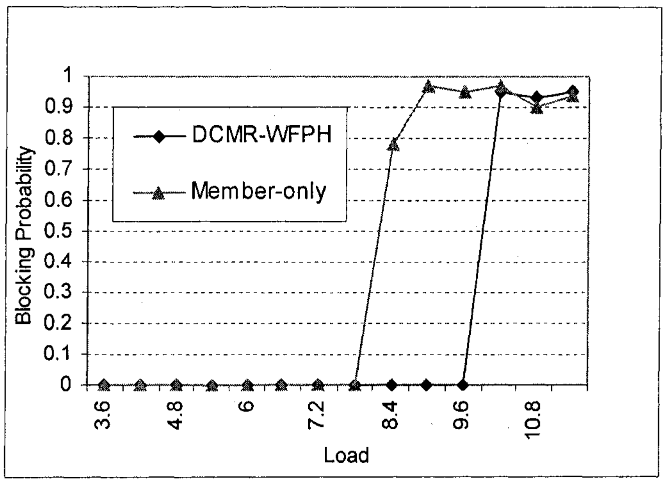


Figure 4.3: Blocking Probability when S=50%.

Figures 4.4 and 4.5 show the overall network utilization for both algorithms, for $S = 30\%$ and $S = 50\%$ respectively. The overall network utilization is an indication of network resources usage under different algorithms and at each load level. It helps to explain why the blocking probabilities saturate: The network wavelength utilization increases significantly at saturation loads, which means forests are starting to occupy much more wavelengths. The data points where the utilization levels rise significantly coincide with the saturation load points. It is apparent from the figures that the network utilization from the Member-only increases at an earlier point than that of the DCMR-WFPH. It means that traffic congestions take place earlier under the Member-only algorithm. The reason for this behavior is because the Shortest-Path-Heuristics based tree generation algorithms do not consider concurrent network resource utilization. Therefore the Member-only algorithm leads to congestions and reaches saturation loads earlier in the dynamic environment.

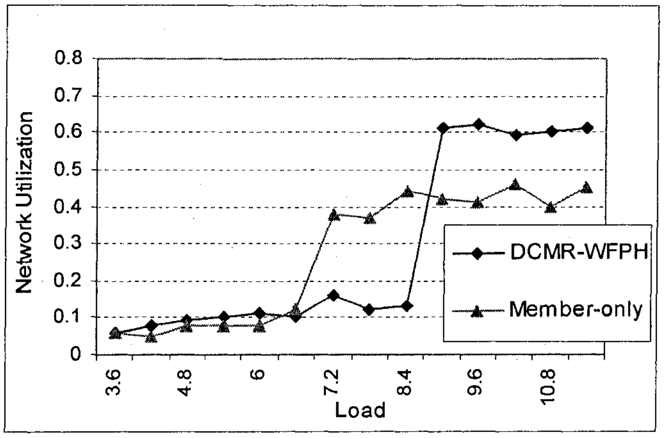


Figure 4.4: Network Utilization when S = 30 %.

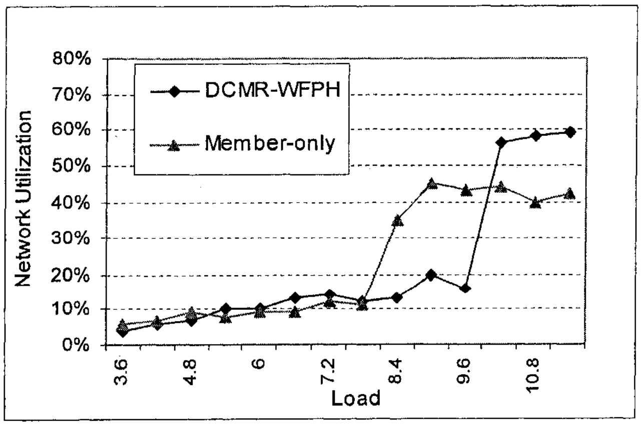


Figure 4.5: Network Utilization when S =50 %.

After the load reaches saturation, the utilization for the Member-only appears lower than that from DCMR-WFPH. This explains why the Member-only performs slightly better than DCMR-WFPH after that point. The reason that DCMR-WFPH utilizes more wavelengths is that it tries to balance the loads leading to longer routes, especially when congestion tends to occur at higher loads. This also means the DCMR-WFPH algorithm utilizes the network resources better in dynamic environment compared to the Member-only—it can accommodate more multicast sessions.

Figures 4.6 and 4.7 present the load balancing performances for both algorithms. It clearly shows that the DCMR-WFPH balances traffic loads much better than the Member-only algorithm. Under the DCMR-WFPH, most of the links have their utilization values close to that of the network overall one. In comparison, Member-only demonstrates a much poorer load balancing performance. Note that the data points beyond the saturation points are more meaningful to our discussion. The reason is before saturation loads, the overall network utilization itself is low (at around 10%).

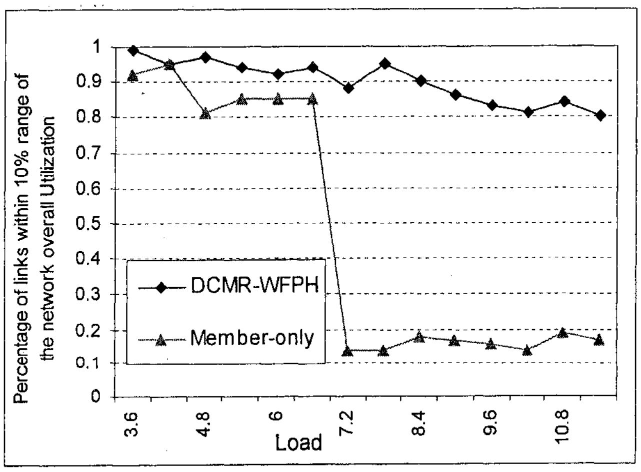


Figure 4.6: Load Balancing performance when S = 30%.

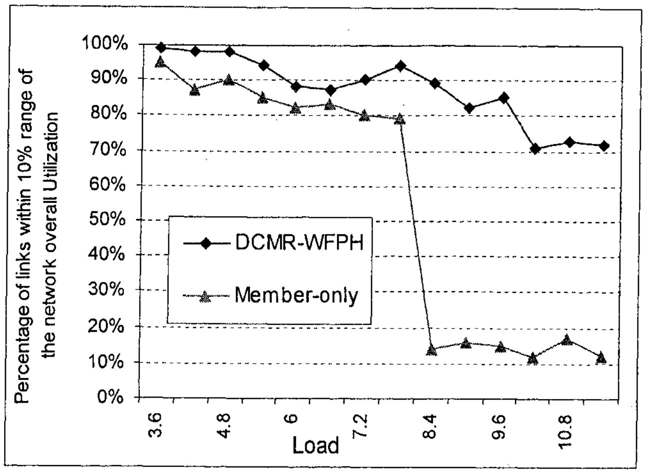


Figure 4.7: Load Balancing performance when S = 50 %.

The figures also suggest that load balancing has an impact on blocking probability behaviors. For example, in Figure 4.6, the load balancing performance drops for the Member-only case at load level 7.2. Its blocking probability rises sharply at that point as well. This means the congestion caused by the bottleneck links have affected the admission of connection requests. The same argument applies to Figure 4.7.

In summary, the simulation data have proved through comparison that the proposed DCMR-WFPH achieved our objectives: It successfully addressed the load balancing aspect in the traffic engineering requirement. It also exhibits a reasonably better blocking probability performance compared to the SPH based Member-only algorithm in dynamic traffic environment.

4.3 Trimming light forests to lower blocking probability

In our simulation results as shown in last section, saturation load serves as an indicator of how well different algorithms are able to accommodate multicast requests at certain load level. If an algorithm exhibits higher saturation load point, it means the network can accept more multicast connection requests under the algorithm. In other words, the algorithm has a better blocking probability performance.

However, the algorithm is only one of the many factors that affect the saturation load. Under the same algorithm, different parameter settings can also affect the saturation load statistics. In a given network, the physical network resources parameters of *S*, *M*, and *W* are fixed. The locations of the *MC* nodes are fixed. Assume the fraction of destinations upon the request arrivals, *T* is also fixed. We further experiment by varying other parameters in the algorithm to observe if the saturation load can be affected. During the previous experiment, when a multicast request arrives, the network tries to accommodate each request with all the available wavelength resources at any given time. The algorithm does not limit the number of wavelengths each forest can occupy. We now introduce another mechanism to *trim the forests*: Set a limit on the number of wavelengths each forest can occupy. We denote this limit as W_F. Trimming the forests prevent individual forests from overly taking up network resources. Thus, we expect the overall blocking probability, thus the saturation point will be reduced via limiting W_F. Table 4.1 tabulates our experiment results for both algorithms under a typical network parameter setting. It shows that the saturation points are the highest when the forests have a W_F of 9. We therefore conclude that trimming light forests can improve the overall network blocking probability behavior. However, we conjecture the exact number of ideal wavelength usage limit is subject to specific network and traffic settings. Further research is needed to determine the ideal limit for forests so that the connection requests can experience a better overall blocking probability performance. In reality, simulation experiments can be utilized to determine an ideal wavelength usage limit or a range of that limit for a specific topology and network condition. The statistics can benefit the network planning so that it can admit more multicast requests.

Table 4.1: The effect of limiting forest wavelengths usage on Saturation Points, assuming S=30%, M=5, T=50%, W=16.

W_F	DCMR-WFPH		Member-only	
	Saturation load	Blocking Probability	Saturation load	Blocking Probability
5	8.8	98%	3.6	88%
6	9.0	94%	5.4	93%
7	8.4	97%	6.0	85%
8	9.0	97%	6.6	88%
9	10.2	96%	7.8	87%
10	8.4	93%	6.6	91%
11	9.0	97%	6.6	88%
12	8.4	96%	4.8	87%
16	9.0	97%	7.2	81%

4.4 The effect of limited network maximum fanout capacity

In the DCMR-WFPH algorithm, the parameter *M* sets the limit of the maximum fanout capacity for all the *MC* nodes in the network. Therefore, for each wavelength channel connection that traverses an *MC* node and that requires splitting, the maximum number of outgoing links supported in the light tree is bound by *M*. This is a realistic assumption from network engineering point of view, since *MC* nodes may only be equipped with certain types of splitters that can only support a limited fanout. For example, the network designer may choose the network maximum fanout capacity to be the average degree among all the network nodes. Light trees therefore can only branch to a certain degree of *M*. On the contrary, Member-only and other sparse-splitting MC-RWA algorithms assume ample maximum fanout capacity, so that the number of outgoing links from *MC* nodes is not limited.

Moreover, note that when *M* is set to be 1, it is equivalent to assuming the network is only equipped with the "Drop and Continue" switches since all nodes can only support one outgoing link. We call this type of networks "Dac-only" networks. Due to concerns such as network evolution, cost, etc., SaD switches may not be equipped in some networks. Light tree services are therefore not supported in "DaC-only" networks. However, optical multicast can still be realized by these "DaC" switches, although multicast destinations will receive signals along light paths instead of light trees. We assume all nodes support the DaC services as stated in the previous chapter. The only difference is that when $M>1$, additional SaD switches are incorporated in the sparsely located *MC* nodes. Note that in "Dac-only" networks, multicast sessions can still utilize multiple wavelengths to form light forests.

In the next set of experiments, we investigate how sparse splitting networks with different maximum fanout capacity perform in a dynamic environment. Simulations are run on the same NSFNet topology and under the DCMR-WFPH algorithm. *M* is set to be 1, 2, and 3 respectively. Again when *M* is set to be 3, all of the *MC* nodes have ample splitting capacity. Two groups of data are presented for the cases where $S = 30\ \%$ and $S = 50\ \%$, respectively. Since the data points significantly overlap, we demonstrate the results using tables. We pay more attentions to the DaC-only cases where *M* is set to be 1.

Table 4.2: The effect of limiting maximum fanout capacity, blocking probability behavior when S=30%, T=50%,W=16.

Load	Blocking Probability		
	M=1	M=2	M=3
11.4	99%	97%	96%
10.8	98%	98%	98%
10.2	99%	96%	97%
9.6	98%	92%	98%
9.0	98%	96%	97%
8.4	0	0	0
7.8	0	0	0
7.2	0	0	0
6.6	0	0	0
6.0	0	0	0
5.4	0	0	0
4.8	0	0	0
4.2	0	0	0
3.6	0	0	0

Table 4.3: The effect of limiting maximum fanout capacity, blocking probability behavior when S=50%, T=50%,W=16.

Load	Blocking Probability		
	M=1	M=2	M=3
11.4	99%	98%	95%
10.8	99%	94%	93%
10.2	99%	96%	95%
9.6	99%	94%	0
9.0	99%	96%	0
8.4	0	0	0
7.8	0	0	0
7.2	0	0	0
6.6	0	0	0
6.0	0	0	0
5.4	0	0	0
4.8	0	0	0
4.2	0	0	0
3.6	0	0	0

Table 4.2 compares the blocking probability performance for different M settings for the case where $S = 30\%$. It shows that the saturation loads are equivalent regardless how M is set. The blocking probabilities are slightly higher for the DaC only networks, after the saturation load level. The results for the case when $S = 50\%$ is presented in Table 4.3,

where it indicates a higher saturation load for $M = 3$. Nonetheless, the difference is not significant. The saturation loads for $M = 1$ and 2 are similar. Therefore, in dynamic traffic environment, depending on the network topologies and conditions, increasing the maximum fanout capacities for sparsely located MC nodes does not always guarantee better blocking probability performances. Certain types of networks that are only equipped with DaC switches can demonstrate similar capacity to accommodate dynamic multicast requests compared to those with SaD switches. However, in general, the performance is dependent on the specific network topologies, as well as the *MC* nodes settings.

Table 4.4: The effect of limiting maximum fanout capacity, Network Utilization data when S=30%, T=50%,W=16.

Load	Overall Network Utilization		
	M=1	M=2	M=3
11.4	69%	56%	61%
10.8	66%	62%	60%
10.2	65%	59%	59%
9.6	62%	55%	62%
9.0	64%	57%	61%
8.4	13%	16%	13%
7.8	15%	12%	12%
7.2	17%	14%	16%
6.6	9%	13%	10%
6.0	10%	11%	11%
5.4	9%	9%	10%
4.8	9%	10%	9%
4.2	8%	9%	8%
3.6	7%	6%	6%

Table 4.5 : The effect of limiting maximum fanout capacity, Network Utilization data when S=50%, T=50%,W=16.

Load	Overall Network Utilization		
	M=1	M=2	M=3
11.4	67%	56%	59%
10.8	66%	56%	58%
10.2	64%	58%	56%
9.6	66%	57%	16%
9.0	68%	58%	20%
8.4	18%	16%	13%
7.8	12%	10%	12%
7.2	12%	15%	14%
6.6	10%	12%	13%
6.0	11%	10%	10%
5.4	9%	9%	10%
4.8	8%	8%	7%
4.2	9%	7%	6%
3.6	6%	8%	4%

Tables 4.4 and 4.5 present the network utilization data for $S = 30\%$ and $S = 50\%$ respectively. We are mainly interested in the data points after the saturation loads are reached. It generally shows that when $M = 1$, the network resources is utilized more compared to the situations when $M > 1$. The reason is because longer routes and more wavelength channels are utilized to reach the multiple destinations when the networks do not have splitting capability.

Load balancing data are shown in Tables 4.6 and 4.7. Again we are mainly concerned with the data points beyond the saturation loads. The interesting phenomenon is that, the "DaC-only" networks exhibit slightly better load balancing behavior. Therefore, it indicates that traffic loads are easier to be balanced with light path services rather than light tree services in sparse-splitting networks. This is also conceivable since the sparsely located *MC* nodes present constraints for how the light trees are produced and branched. Therefore the load balancing performance is affected.

Table 4.6: The effect of limiting maximum fanout capacity, load balancing data when S=30%, T=50%,W=16.

Load	Percentage of links within +/-10% of the overall Network		
	M=1	M=2	M=3
11.4	79%	60%	69%
10.8	94%	76%	84%
10.2	90%	90%	73%
9.6	74%	74%	83%
9.0	90%	89%	86%
8.4	83%	88%	90%
7.8	88%	90%	95%
7.2	84%	88%	88%
6.6	90%	91%	94%
6.0	92%	96%	92%
5.4	94%	94%	94%
4.8	93%	91%	97%
4.2	97%	96%	95%
3.6	98%	99%	99%

Table 4.7: The effect of limiting maximum fanout capacity, load balancing data when S=50%, T=50%,W=16.

Load	Percentage of links within +/-10% of the overall Network		
	M=1	M=2	M=3
11.4	81%	76%	72%
10.8	83%	57%	73%
10.2	86%	76%	69%
9.6	90%	76%	85%
9.0	83%	75%	82%
8.4	97%	97%	89%
7.8	94%	96%	94%
7.2	93%	85%	90%
6.6	93%	92%	87%
6.0	91%	95%	88%
5.4	96%	90%	94%
4.8	97%	98%	98%
4.2	95%	98%	98%
3.6	98%	96%	99%

4.5 Summary

This chapter has presented the simulation results. Through comparison with the Member-only in dynamic traffic patterns, the DCMR-WFPH algorithm has demonstrated advantages in load balancing, as well as blocking probability performance metrics. In particular, the networks under the DCMR-WFPH algorithm reached saturation loads later, with most of the links having their utilization close to that from the overall network. Further experiments have showed that we can apply the trimming forest method to the algorithms so that better blocking probability performance can be achieved. This method has utilized the fact that the algorithms produce light forests, so that the number of wavelengths they occupy can be varied. The effects on limiting network maximum fanout capacities have also been shown via simulation experiments. The interesting discovery is that some DaC-only sparse-splitting networks can achieve similar performances to those equipped with SaD switches.

Chapter 5 Ensuring loop-free operations in light tree generation

This chapter provides formal analysis on the main operations in the DCMR-WFPH algorithm. In particular, it is proven that the algorithm indeed produce *trees*. By definition, trees are *connected* and *acyclic* graphs [Wes96]. The terms of loop and cycle are used interchangeably in the context. By contrast, most other algorithms in the literature assume they can produce trees successfully. Little such formal justifications has been made to previous tree-building algorithms.

5.1 The possibility of producing loops in tree generation process

Routing loops are undesirable structures in routing process. Loops will be produced if cares are not taken. One source of the loop generation is an inconsistent maintenance of the network state information database. Different nodes may contain different versions of the network topology and resources information. Normal routing algorithm operation with inconsistent network state information can lead to loops in the routing results. This is particularly the case for distributive routing algorithms since every node makes their independent routing decisions. Source routing algorithms on the other hand, can detect loops by an ordered scanning of the candidate paths. This scenario applies to both the unicast and multicast routing algorithms. For example, loops frequently occur in some existing IP multicast protocol such as DVMRP. It was noted that in Optical networks, loops can cause serious problems so that they should be avoided as much as possible in multicast routing algorithms [Rou03].

Since distance is the deciding metric for shortest path heuristics, they have an intrinsic property that can be utilized to avoid loops since paths containing loops will be inevitably longer than the ones that do not involve loops. However, such is not the case in distributive widest path heuristics, where width is the deciding metric. A path with a loop and a path without a loop may have the same width if the loop does not involve the bottleneck links. Therefore, loops can be formed in the routing process and it is not easy to detect them.

Nonetheless, it has been proven in [Wan96] that the proposed shortest-widest routing algorithm can guarantee loop-free in distributed computations. The Widest-First-Path Heuristics we adopt in the DCMR-WFPH algorithm is based on the loop-free shortest-widest algorithm. Therefore the paths we use to construct trees will not contain any loops.

Most of the loop prevention and detection methods so far focused on the issue of loop formations along paths. However, if cares are not taken, loops can also be formed during the tree generation process.

Figure 5.1 illustrates such a general looping scenario in the light tree generation process. In the figure, the node s is the source node for the multicast session and d_1, d_2 and d_3 are destinations.

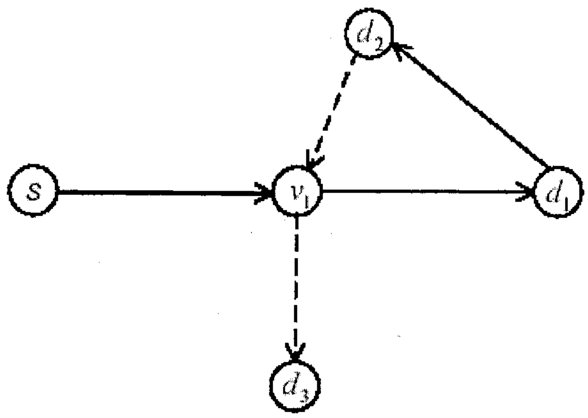


Figure 5.1: Possible loop scenario in tree generation process

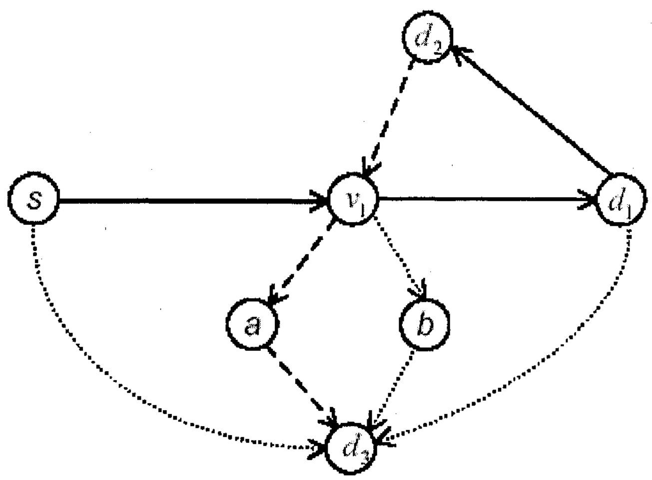


Figure 5.2: Possible loop scenario in tree generation process from algorithms based on widest path heuristics

DCMR-WFPH algorithm is a distributive dynamic multicast routing algorithm that is based on a widest-first-path-heuristics. As we pointed out, the special property of the widest path routing algorithm makes it easier for loops to form. This possibility lies in both path generation as well as tree generation process. Figure 5.2 shows a typical loop formation scenario during the tree generation process with algorithms using widest-first-path-heuristics. Again, in the figure, node s is the source node, d_1, d_2 and d_3 are destinations.

Suppose current sub-tree already consists of the paths $P(s, d_1)$ and $P(d_1, d_2)$. Node v_1 is a node on the path $P(s, d_1)$ and $v_1 \in V_T$. That is to say, v_1 is a node that allows supporting new branches. Node d_3 is the next destination node waiting to join the tree. Due to the sparse-splitting constraints, v_1 can only connect to nodes that allow new branches, i.e., nodes in the current set of V_T. The principle is to select the widest and then shortest feasible path from the candidate paths. Currently the set of V_T includes nodes $\{s, d_1, d_2, v_1\}$. Four widest candidate paths, namely, $P(s, d_3)$, $P(d_1, d_3)$, $P(d_2, v_1, a, d_3)$ and $P(v_1, b, d_3)$ are generated for d_3. Note that between the nodes v_1 and d_3, $P(v_1, b, d_3)$ is chosen over $P(v_1, a, d_3)$ because of its better width attribute. Following the candidate paths generation stage is the path validation stage. It is possible that paths $P(s, d_3)$, $P(d_1, d_3)$, and $P(v_1, b, d_3)$ all turn out to be infeasible since they traverse some nodes in the set of V_T', i.e., nodes that do not allow new branches. As a result, path $P(d_2, v_1, a, d_3)$ is selected for the destination member d_3. However, as shown in the figure, the effect is a loop is produced after the path $P(d_2, v_1, a, d_3)$ is added to the sub-tree.

The DCMR-WFPH algorithm was carefully designed so that loops will not occur during the tree generation process. We provide the detailed analysis in the following section.

5.2 Properties of the DCMR-WFPH algorithm

In this section we introduce and justify a series of formal statements. They describe some of the primary properties of the DCMR-WFPH tree generation operation. The most important feature, as we have pointed out, is that the algorithm guarantees that it does not produce any loops during the tree generation process.

One of the primary operations in the DCMR-WFPH algorithm is the Path Construction Utility. Therefore we concentrate our analysis on that procedure.

Based on the specifications of the PCU and the principle of the DCMR-WFPH algorithm, we reach the following series of lemmas and theorems.

Lemma 5.1: During each individual tree generation process, consider the subgraph in transition, denoted by $G'(N',E')$, that is already present before the start of the next PCU operation for the next new path *P*, and the corresponding sets of V_T and V_T' at that moment. In G', N' is the set of nodes and E' is the set of all the edges. We have the following arguments: $V_T, V_T' \subseteq N'$, $V_T \bigcup V_T' = N', V_T \bigcap V_T' = \phi$.

Proof: Note that the subgraph $G'(N',E')$ is a transient graph: It keeps expanding throughout each sub-tree's formation process. A single PCU operation attaches one new path to the existing G' at a time. For the purpose of illustration, the notation of $G'(N',E')$ in the context applies to any instance of the subgraph that exists before the next new path is constructed. The following is the proof of this lemma.

For any node that belongs to either V_T or V_T', it must belong to N'. Algorithm DCMR-WFPH classifies nodes into the set of V_T or V_T'. It is performed to the nodes along the paths after they have been produced. These paths are components of the existing subgraph G'. Therefore, we have $V_T, V'_T \subseteq N'$. Note that as a special case, each time when a tree construction stage initially starts, $G'(N', E')$ only has one element—the source node s, with $V_T = \{s\} = N'$, and $V'_T = \phi$, and the statement is obviously true for that case.

The statement above already implies $V_T \cup V'_T \subseteq N'$. On the other hand, since for every node on the paths that have been generated, thus on the current subgraph G', DCMR-WFPH algorithm will classify it into either V_T or V'_T from Step 4.2.4 to Step 4.2.6, which means $N' \subseteq V_T \cup V'_T$. Therefore, we have $V_T \cup V'_T = N'$.

According to the definition of V_T and V'_T, they are mutually exclusive thus it concludes that $V_T \cap V'_T = \phi$. ■

Lemma 5.1 indicates that during the tree generation process, all of the nodes that are included will be analyzed and classified according to their current splitting capacities. This process ensures that the splitting constraints will always be respected during the algorithm operations.

Lemma 5.2: Each constructed path *P(v,u)* contains at least two distinct nodes *v* and *u*, with $v \in V_T$ and $u \in D^*$, where $D^* \cap V_T = \phi$.

Proof: Each successfully constructed path *P(v,u)* contains at least two nodes $v \in V_T$ and $u \in D^*$ as required by the PCU. Since Step 4.2.7 of DCMR-WFPH checks and excludes from D^* any node that is already on G', therefore $D^* \cap N' = \phi$. It then follows from Lemma 5.1 that $D^* \cap V_T = \phi$. Therefore, the lemma holds. ■

Lemma 5.2 means that PCU does produce paths that at least contain two different nodes. Furthermore, it ensures every time when the PCU is invoked, it produces a new path that leads to a destination member that is in the waiting list and that is not included by the existing subgraph. In contrast, many multicast routing algorithms, including the Member-only algorithm do not provide this guarantee, such that a destination member node may reside on the existing subgraph.

Lemma 5.3: During each tree formation, consider the existing subgraph $G'(N', E')$, before the start of the next PCU operation for the new path *P*, and the corresponding set of V_T at that moment. Upon the successful construction of *P*, it contains one and only one vertex that is already included in G', which serves as the source end point for the new path *P* and belongs to V_T .

Proof: We prove this Lemma by induction.

The algorithm defines that each tree is constructed by concatenating paths one at a time. Denote them as paths P_1, P_2, P_3, ..., P_N, where N is an arbitrary finite positive integer. As the base case of the induction proof and a special case, before the first path P_1 is generated, G' only contains one node—the source node *s* and contains no edges. The node *s*, which is the source of the whole multicast session, always belongs to the set of V_T as stated in Step 2 of the DCMR-WFPH. It also serves as the source end point for P_1 as required by Step 3 in the DCMR-WFPH for all candidate paths in PCU. Therefore, P_1 contains one and only one vertex that is already included in G', which serves as the source endpoint for it and belongs to V_T. Note that the first path P_1 will always be returned successfully upon completion of the PCU since $V_T' = \phi$, although it may be invalidated by the width checking procedure later on in Step 4.1, which is a separate matter.

We assume the argument given by Lemma 5.3 holds for path P_k, $1 \le k < N$.

Consider the case for path P_{k+1}. At this instant, there exists a subgraph $G'(N', E')$, as well as sets V_T and V_T', that are already present before P_{k+1} is produced.

Again, Step A of the PCU requires that the source endpoints of all the candidate paths for P_{k+1} are all coming from the node pool of the current set of V_T. Therefore, upon the completion of PCU, P_{k+1} will contain at least one vertex from V_T, which serves as the source endpoint for P_{k+1}. According to Lemma 5.1, the vertex is also included in the existing subgraph G'. Therefore, P_{k+1} contains one vertex that is already included in G', which serves as the source endpoint for the new path P_{k+1} and belongs to V_T.

Now we prove P_{k+1} contains no more than one vertex from G' by contradiction.

Suppose P_{k+1} contains more than one vertex from G', i.e., from N'. From Lemma 5.1, it follows that the nodes from N' would belong to either V_T or V_T'. It is not possible for the nodes to belong to V_T' since the candidate paths containing nodes from V_T' will be invalidated by Step C.1.1 of PCU. Therefore, the only possibility is P_{k+1} contains more than one vertex from the set of V_T. However, by backtracking the candidate paths, Step C.1.2 of PCU guarantees it will return a valid path on the first encounter of any node from V_T. From Lemmas 5.1 and 5.2 we know the destination $u \in D^*$ for path P_{k+1} will not belong to G', thus neither V_T. We then conclude that the assumption is not true, so that it is impossible for P_{k+1} to contain more than one vertex from G'.

Hence, the argument holds for the case of P_{k+1}, which completes the induction proof of Lemma 5.3. ■

Lemma 5.3 illustrates an important property of the tree expansion process for the DCMR-WFPH algorithm. Each time when a new path is generated, it connects from only one node that allows new branches on the existing subgraph to a new destination that is waiting to join the tree. No other nodes on the existing subgraph will be involved.

Theorem 5.1: The graphs produced by DCMR-WFPH algorithm do not contain any loop, or *cycle* in a graph theory term.

Proof: During the proof of this theorem, we define *route* as a *walk* with no repeated vertex and a *cycle* is a closed *trail* where "first = last" is the only vertex repetition. Walk and trail are conventional terms as defined in [Wes96]. In this proof, we change the original graph theory term *path* into "route" to avoid confusion. Thus, the route and cycle here can be results of multiple processes of PCU, although a path in our algorithm is also a route produced by a single process of PCU.

In the algorithm, each graph is formed by expanding G' (as defined in Lemma 5.1). At the start, G' consists of only one single node—the Source node of the multicast session. One path is added to G' each time the PCU is completed. It suffices to show that at any given instance of time, G' does not contain any cycle. We prove the argument by contradiction. Figure 5.3 depicts the scenario described in this proof.

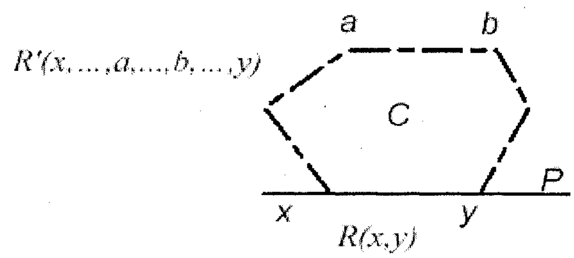


Figure 5.3: Assume G' contains a cycle.

Suppose there exists an instance of G', such that G' contains at least one cycle. Randomly select any of these cycles and denote it by *C*.

From Lemma 5.2, we know that each constructed path contains at least two distinct nodes. The graph G', therefore *C*, is formed by adding one path at a time, although *C* may contain "partial" paths. That is to say, every edge in *C* belongs to some path. Then we can find from *C*, at least a pair of distinct nodes, say *(x,y)*, that is added to *C*, thus G', during a single path construction. Denote that path by *P*. Therefore, the route *R(x,y)* resides in *C* as part of *P*, as depicted in Figure 5.3.

From graph theory, we know that for any pair of distinct nodes on a cycle, there must be at least two distinct routes on the cycle between this pair of nodes [Wes96]. Select the node pair of *(x,y)* as such a pair of distinct nodes on the cycle *C*. Then there must be at least another route, say *R'(x,...,a,...,b,...,y)* that is distinct from *R(x,y)*, where *R'(x,...,a,...,b,...,y)* also belongs to *C*, thus G'. Therefore, the route *R'(x,...,a,...,b,...,y)* as part of G' already exists before the construction of *P* starts. Hence, it follows that upon the successful construction of *P*, *P* includes distinct nodes *x* and *y*, that already reside in the previously existing G'. This contradicts the result of Lemma 5.3 presented above. Therefore, the assumption does not hold.

This completes the proof of Theorem 5.1. ■

Theorem 5.2: Each time when a tree generation stage in the DCMR-WFPH algorithm terminates, G' is guaranteed to be a tree.

Proof: A tree is defined as a connected and acyclic graph [Wes96]. In each tree generation process, G' expands by adding one path at a time utilizing the PCU, starting from the multicast source node. Lemma 5.3 implies that G' will remain connected throughout the process. Theorem 5.1 guarantees that G' is acyclic at any instance. Therefore, G' is always a tree at any instance. ■

Theorem 5.1 states that all of the subgraphs produced by the routing algorithm do not contain any cycles. Following that, Theorem 5.2 justifies that DCMR-WFPH indeed produce trees. As far as the final light forest is concerned, after the wavelength assignment procedure succeeds, the trees will either be on different wavelengths, or be link-disjoint in the case that they do utilize the same wavelength. In either case, the trees do not interfere with each other thus do not cause any problems.

Beyond guaranteeing loop-less, PCU operation also preserves the QoS and traffic engineering objectives, as shown by the following lemma.

Lemma 5.4: The successfully returned path $P(x, u)$ by Step C.1.2 of PCU is at least as wide as, and as short as $P(v, u)$.

Proof: We have $x \neq u$ as implied by Lemma 5.2. In the case that $x = v$, the width and length of $P(x, u)$ are exactly the same as that of $P(v, u)$.

In the case that $x \neq v$, then the original path $P(v, u)$ consists of two segment paths— $P(v, x)$ and $P(x, u)$, where the part of $P(v, x)$ is cut off from $P(v, u)$ to return $P(x, u)$. By definition, the width of path $P(v, u)$ is given by the link(s) that has minimum width. Consider the path of $P(x, u)$. Suppose that $B_{P(x,u)} < B_{P(v,u)}$, it then implies that the segment of *P(x, u)* contains the link with minimum width. Then $B_{P(v,u)}$ would be governed by the smaller $B_{P(x,u)}$ instead of the original $B_{P(v,u)}$, since *P(x, u)* is part of *P(v, u)*. Therefore, it is impossible that $B_{P(x,u)} < B_{P(v,u)}$, which means $B_{P(x,u)} \geq B_{P(v,u)}$. It is also obvious that *P(x, u)* is shorter than *P(v, u)* in this case.

That completes the proof of Lemma 5.4. ■

Lemma 5.4 proves that the PCU is also designed to preserve the algorithm's QoS and traffic engineering objectives: It achieves that by producing paths that are widest and shortest from the available candidates.

5.3 Summary

While Chapter 4 has demonstrated the advantages of the algorithm in the dynamic environment, the analysis in this chapter focuses more on the constraint and multicast aspects of the DC-MCRWA problem.

It has been proven that the sparse splitting and the splitting capacity constraints are always respected during the algorithm operation; and that each process ensures that the subgraph under construction only expands by including new destinations. By selecting the widest and shortest path from the candidates, the PCU also preserves the "widest first and distance second" principle, thus contributing to the QoS and traffic engineering objectives of the algorithm. The analysis has successfully proven that the DCMR-WFPH is guaranteed to generate trees, i.e., no loops will be produced during the algorithm operations.

Chapter 6 Conclusions and future research

This chapter draws conclusions of this thesis work, as well as outlines some areas for future research.

6.1 Conclusions

In a sparse-splitting WDM networks, multicast routing and wavelength assignment operations are constrained. While all of the previous research work in this area has assumed a static traffic environment, this research work provides an investigation into the dynamic traffic scenario.

A survey of recent advances in optical multicast is presented in Chapter 2. We focused on works in the area of multicast routing and wavelength assignment in sparse-splitting networks.

The problem of dynamic constrained multicast routing and wavelength assignment (DC-MCRWA) has been raised and discussed in detail in Chapter 3. The analytical approach presented in this chapter has combined the aspects of dynamic routing and constrained multicast and wavelength assignment. Under the objectives of load balancing and bandwidth requirement, we have proposed a dynamic constrained multicast routing algorithm (DCMR-WPFH) that is based on a widest-path-first heuristics. This is a contrary to all of the other multicast routing and wavelength assignment algorithms known to date, which mostly incorporate the Shortest-Path-based Heuristics in their multicast tree generation process.

"Width first and distance second" is the principle concept behind the design of the DCMR-WPFH algorithm. The main path generation process, i.e., PCU, utilizes this principle. Destination members are ordered according to the width first criterion. Candidate paths for each member are arranged based on their width first. Feasible paths are constructed to ensure they are as wide and short as possible.

The width metric is also utilized during the tree generation stage so that the algorithm can construct load-balanced trees. Furthermore, the width metric contributes in the algorithm to ensure that the bandwidth requirement for the whole multicast session can be fulfilled.

Destination members that are already included in the existing trees are excluded from further consideration so that the algorithm is much more efficient than other approaches. This process also helps prevent loops in the routing results.

A Bottleneck wavelength assignment approach is introduced in the wavelength assignment procedure. Instead of having to search the entire wavelength space, only wavelengths available on bottleneck links are examined for wavelength assignment purpose. This approach is uniquely available and suitable for dynamic wavelength assignment procedures.

The advantages of the design principle have been successfully proven by simulation results in Chapter 4. Compared to the Member-only algorithm, a well known shortest-path-heuristics based tree generation algorithm, DCMR-WPFH outperforms significantly in terms of load balancing behavior. It also demonstrates better capacity in admitting multicast requests under the dynamic traffic pattern—indicated by a higher saturation load.

By setting a limit on the number of wavelengths each forest can occupy—a method we refer to as "trimming the forest", blocking probability performance can be further improved. This phenomenon is also observed through simulation experiment.

We further investigated the effect on limiting the maximum fanout capacity on the sparse-splitting networks. Simulation results indicate that, depending on the network topologies and conditions, networks equipped with only DaC switches can also perform reasonably well compared to those with ample light splitting capacities.

The scenario of loop formation during the tree generation process is studied in Chapter 5. Routing algorithms that are based on width metrics are prone to loop formations due to its special route selection criterion, no matter in path or tree generation process. We have formally justified a series of statements regarding the primary operations in the DCMR-WPFH algorithm. In particular, we have proved that the algorithm is guaranteed to generate trees. Moreover, the path selection process preserved the principle objectives of the algorithm by selecting paths that are as wide and short as possible.

6.2 Areas of future research

Throughout the thesis, a number of areas that warrant future research efforts have been identified. We summarize them as follows.

1) The multiple routing metric type issue: As a single metric type routing approach, the Shortest-path-heuristics has been dominantly utilized and performs reasonably well in most of the situations. However, there are circumstances where other routing metric types can be incorporated in order to achieve a variety of routing objectives. The work as shown in this thesis presents such an example. However, routing protocols utilizing multiple metric types still suffer from tradeoffs. For example, we have shown in Chapter 3 that the single mixed typed routing approach presented in [Ass01] may only be able to achieve a sub-optimal outcome in certain circumstances.

   Another counterexample has shown that the famous shortest-widest routing algorithm proposed in [Wan96] in fact can only guarantee that a widest path will be produced between a pair of nodes. It has been conjectured that finding an absolute widest and also shortest path between a pair of nodes is an NP-complete problem. More detailed formal analysis on this problem, as well as further investigation into the multiple metric types routing issue are left to future research.

2) The number of wavelengths that each forest should occupy in order to achieve a best blocking probability performance: It has been shown that limiting the number of wavelengths that each forest occupies can affect the blocking probability performance. The conjecture is that the exact number of wavelengths that can help achieve the best performance is influenced by a number of factors including network and traffic conditions. More research could

be dedicated to this topic to establish the relationship between the performance and the factors.

3) The effect of limiting network maximum fanout capacity: Simulation experiments indicates that some networks that only equipped with DaC switches can achieve a reasonably well dynamic performance compared to those with SaD switches. Further researches can be done to identify the conditions and situations where the maximum fanout capacity does or does not affect the performance significantly.

Bibliography

[Ali00a] M. Ali and J. Deogun, "Power-efficient Design of Multicast Wavelength-Routed Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 10, Oct. 2000, pp. 1852-1862.

[Ali00b] M. Ali and J. S. Deogun, "Cost-effective implementation of multicasting in wavelength routed networks," *IEEE Journal of Lightwave Technology*, Vol. 18, No. 12, Dec. 2000, pp. 1628-1638.

[Ali00c] M. Ali and J. Deogun, "Allocation of Splitting Nodes in Wavelength-routed Networks," *Photonic Network Communication*, Vol. 2, No. 3, Aug. 2000, pp. 245–263.

[Ass01] C. Assi, M. Ali, R. Kurtz, and D. Guo. "Optical networking and real-time provisioning: An integrated vision for the next-generation internet," *IEEE Network*, Vol. 15, No. 4, Jul./Aug. 2001, pp. 36-45.

[Ban01] A. Banerjee, et al., "Generalized Multiprotocol Label Switching: An overview of routing and management enhancements," *IEEE Communications Magazine*, Vol. 39, No. 1, Jan. 2001, pp 144-150.

[Bau95] F. Bauer and A. Varma, "Degree-constrained multicasting in point-to-point networks," *Proceedings of IEEE INFOCOM'95*, pp. 369-376.

[Bor95] M. S. Borella and B. Mukherjee, "A Reservation-Base Multicasting Protocol for WDM Local Lightwave Networks," *Proceedings of IEEE International Conference on Communications (ICC)*, 1995, pp. 1277-1281.

[Bow04] C. Boworntummarat, L. Wuttisittikulkij, and S. Segkhoonthod, "Light-tree based protection strategies for multicast traffic in transport WDM mesh networks with multifiber systems," *Proceedings of IEEE International Conference on Communications (ICC)*, 2004, Vol. 3, pp. 1791 – 1795.

[Cao05] Y. Cao and O. Yu, "Optimal placement of light splitters and wavelength converters for multicast in WDM networks," *Proceedings of IEEE International Conference on Communications (ICC)*, 2005, Vol. 1. pp. 585-589.

[Cao05a] Y. Cao and O. Yu, "On the study of group multicast in WDM networks," *Proceedings of IEEE International Conference on Communications (ICC),* 2005, Vol. 3, pp. 1625– 1630.

[Cao05b] Y. Cao and O. Yu, "QoS-guaranteed routing and wavelength assignment for group multicast in optical WDM networks," *Proceedings of 2005 Conference on Optical Network Design and Modeling*, Feb. 2005, pp. 175 – 184.

[Cao06] Y. Cao and O. Yu, "Optimization of loss-balanced multicast in all-optical WDM networks", *Journal of Combinatorial Optimization*, Vol.12, No. 1-2, Sept. 2006, pp.71-82.

[Car97] K. Carlberg and J. Crowcroft, "Building Shared Trees using a One-to-Many Joining Mechanism", *ACM Computer Communication Review*, Jan. 1997, pp. 5-11.

[Cha94] K. Chan and T. P. Yum, "Analysis of Least Congested Path Routing in WDM lightwave Networks," *Proceedings of IEEE INFOCOM'94*, Vol. 2, pp. 962-969.

[Che01] B. Chen and J. Wang, "Constrained wavelength assignment for multicast in WDM networks," *Proceedings of Tenth International Conference on Computer Communications and Networks*, Oct. 2001, pp. 388 – 394.

[Che02] B. Chen and J. Wang, "Efficient routing and wavelength assignment for multicast in WDM networks," *IEEE Journal on Selected Areas in Communications*, Vol. 20, Jan. 2002, pp. 97-109.

[Chl92] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications, An approach to High-Bandwidth Optical WANs," *IEEE Transactions on Communications*, Vol. 40, No. 7, Jul. 1992, pp. 1171-1182.

[De06] T. De and S. Sen, "Multicast Routing and Wavelength Assignment in Sparse Splitting All Optical Networks," *IFIP International Conference on Wireless and Optical Communication Networks*, Apr. 2006, pp. 1-5.

[Den06] Y. Deng and T. T. Lee, "Crosstalk-free conjugate networks for optical multicast switching," *IEEE Journal of Lightwave Technology*, Vol. 24, No. 10, Oct. 2006, pp. 3635-3645.

[Din02] A. Ding, G. S. Poo and S. T. Tan, "An expanded graph model for MCRWA problem in WDM networks," *Proceedings of 27th Annual IEEE Conference on Local Computer Networks (LCN),* Nov. 2002, pp. 557 – 564.

[Din03] A. Ding, and G.S. Poo, "A survey of optical multicast over WDM networks", *Elsevier Computer Communications*, 26 (2003), pp. 193-200.

[Doa93] M. Doar, I. Leslie, "How bad is naïve multicast routing?" *Proceedings of IEEE INFOCOM'93*, pp. 82-89.

[Dut00] R. Dutta and G. N. Rouskas, "A Survey of virtual Topology Design Algorithms for wavelength Routed Optical Networks," *Optical networks*, Vol. 1, No. 1, Jan. 2000, pp. 73-89.

[Esh04] A. E. Eshoul and H. T. Mouftah, "IFF: a novel wavelength assignment scheme for WDM optical networks," *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering*, Vol. 2, May 2004, pp. 965-969.

[Gar77] M. R. Garey, R. L, Graham and D. S. Johnson, "The complexity of computing Steiner minimal trees," *SIAM Journal of Applied Mathematics*, Vol. 32, No. 4, Jun. 1977, pp. 835-859.

[Gar03] P. Garcia, A. Zsigri and A. Guitton, "A multicast reinforcement learning algorithm for WDM optical networks," *Proceedings of the 7th International Conference on Telecommunications*, 2003, (ConTEL 2003), Vol. 2, 11-13 Jun. 2003, pp. 419 – 426.

[Gro04] W. D. Grover, "Mesh-based survivable networks: Options and strategies for Optical, MPLS, SONET, and ATM Networking," *Prentice Hall PTR*, 2004.

[Hak71] S. L. Hakimi, "Steiner's problem in graphs and its implications," *Networks*, 1: 113-133, 1971.

[Ham05] A. M. Hamad and A.E. Kamal, "Routing and wavelength assignment with power aware multicasting in WDM networks," *2nd International Conference on Broadband Networks*, Oct. 2005, Vol. 1, pp. 31 – 40.

[He01] J. He, S. H. Chan, and D. H. K. Tsang, "Routing and Wavelength assignment for WDM multicast networks," *Proceedings of IEEE Global Telecommunications Conference* (GLOBECOM'01), Vol. 3, pp. 1536-1540.

[Ho02] P. H. Ho and H. T. Mouftah, "A novel distributed control protocol in dynamic wavelength-routed optical networks," *IEEE Communications Magazine*, Vol. 40, No. 11, Nov. 2002, pp. 38-45.

[Hu98] W. S. Hu and Q. J. Zeng, "Multicast Optical Cross Connects employing splitter-and-delivery switch," *IEEE Photonic Technical Letter*, Vol. 10, Jul. 1998, pp. 970-972.

[Hu02] X. –D. Hu and M. –H. Zhang, "Multicast routing and wavelength assignment under multi-drop model in WDM networks," *Proceedings of IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*, 2002, (TENCON '02), Vol. 2, Oct. 2002, pp. 1201 – 1204.

[Hu04] X.-D Hu, T. -P Shuai, X. Jia and M. Zhang, "Multicast routing and wavelength assignment in WDM networks with limited drop-offs," *Proceedings of IEEE INFOCOM'04*, Vol. 1, pp. 487-494.

[Hsi03] C. Y. Hsieh and W. Liao, "All Optical Multicast Routing in Sparse-Splitting Optical Networks," *Proceedings of 28th Annual IEEE International Conference on Local Computer Networks*, Oct. 2003, pp. 162-167.

[Hwa92] F. K. Hwang, D. S. Richards, P. Winter, "The Steiner Tree Problem," *Elsevier Science Publishers*, Amsterdam, 1992

[Jia01] X. H. Jia, et al., "Optimization of Wavelength Assignment for QoS multicast in WDM networks," *IEEE Transaction of Communications*, Vol. 49, No. 2, Feb. 2001, pp. 341-350.

[Jia02] X. Jia, X. Hu, L. Ruan, and J. Sun, "Multicast routing, load balancing, and wavelength assignment on tree of rings" *IEEE Communication Letter*, Vol. 6, Feb. 2002, pp. 79-81.

[Jue00] J. P. Jue and G. Xiao, "An adaptive routing algorithm for wavelength-routed optical networks with a distributed control scheme," *Proceedings of the Ninth*

International Conference on Computer Communications and Networks, Oct. 2000, pp. 192-197.

[Kam02] A. E. Kamal and A. K. Al-Yatama, "Blocking Probabilities in Circuit-switched Wavelength Division Multiplexing Networks Under Multicast Service," *Performance Evaluation*, Vol. 47, No. 1, 2002, pp. 43-71.

[Kha04] A. Khalil, C. Assi, A. Hadjiantonis, G. Ellinas, N. Abdellatif and M. Ali, "A Multicast traffic grooming in WDM networks," *2004 Canadian Conference on Electrical and Computer Engineering*, Vol. 2, May 2004, pp. 785 – 788.

[Kou81] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," *Acta Informatica*, Vol 15, 1981, pp. 141-145.

[Lee04] P. Leelarusmee, C. Boworntummarat and L. Wuttisittikulkij, "Design and analysis of five protection schemes for preplanned recovery in multicast WDM networks," *2004 IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, Apr. 2004, pp. 167 – 170.

[Lee05] S. C. Lee, R. Varrazza, O. Ansell and S. Yu, "Highly Flexible 4 × 4 optical crosspoint packet switch matrix for optical multicast operations," *IEEE Photonic Technology Letters*, Vol. 17, No. 4, Apr. 2005, pp. 911-913.

[Lee06] S. C. Lee, R. Varrazza, and S. Yu, "Advanced optical packet switching functions using active vertical-couplers-based optical switch matrix," *IEEE Journal of Selected Topics in Quantum Electronics*, Vol.12, No. 4, Jul./Aug. 2006, pp. 817-827.

[Li99] L. Li, and A. K. Somani, "Dynamic Wavelength Routing Using Congestion and Neighborhood Information," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 5, Oct. 1999, pp. 779-786.

[Li00] D. Li, X. Du, X. Hu, R. Lu and X. Jia, "Minimizing Number of Wavelengths in Multicast Routing Trees in WDM Networks," *Networks*, Vol. 35, No. 4, Jul. 2000, pp. 260-265.

[Lia98] W. Liang and H. Shen, "Multicasting and broadcasting in Large WDM networks," *Proceedings of 1st merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing (IPPS/SPDP)*, 1998, pp. 365-369.

[Lib02] R. Libeskind-Hadas and R. Melhem, "Multicast routing and wavelength assignment in multihop optical networks," *IEEE/ACM Transactions on Networking*, Vol. 10, No. 5, Oct. 2002, pp. 621 – 629.

[Liu06] X. Liu, H. Wang and Y. Li, "Serial Multicast Mode in All-Optical Networks," *IEEE Photonics Technology Letters*, Vol. 18, No. 22, Nov. 2006, pp. 2416 – 2418.

[Low97] E. D. Lowe and D. K. Hunter, "Performance of dynamic path optical networks," *IEE Proceedings of Optoelectronics*, Vol. 144, No. 4, Aug. 1997, pp. 235-239.

[Low00] C. P. Low and Y. J. Lee, "Distributed Multicast Routing with End-to-end Delay and Delay Variation Constraints," *Elsevier Computer Communications*, Vol. 23, No. 9, 2000, pp. 848-862.

[Mod99] E. Modiano, "Random algorithms for scheduling multicast traffic in WDM broadcast-and-select networks," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 3, Jun. 1999, pp. 425-434.

[Moy94] J. Moy, "Multicast routing extensions for OSPF", *Communications of the ACM*, Vol. 37, No. 8, Aug. 1994, pp. 61-66.

[Muk97] B. Mukherjee, "Optical Communication Networks," *McGraw-Hill Publisher*, New York, 1997

[Muk00] B. Mukherjee, "WDM optical communication networks: progress and challenges," *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 10, Oct. 2000, pp. 1810-1824.

[Ort97] Z. Ortiz, G. N. Rouskas, H. 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,* Vol. 27, No. 4, Oct. 1997, pp. 301-310 .

[Pal05] F. Palmieri and U. Fiore, "Implementing effective optical multicast with minimum constrained routing and wavelength assignment," *WSEAS Transactions on Communications*, Vol. 4, No. 7, Jul. 2005, pp. 382-390.

[Poo05] G. –S. Poo and Y. Zhou, "Optical multicast over wavelength-routed WDM networks: A survey," *Elsevier Optical Switching and Networking*, Vol. 2, No. 3, Nov. 2005, pp.176-197.

[Poo06] G. –S. Poo and Y. Zhou, "A New Multicast Wavelength Assignment Algorithm in Wavelength-Routed WDM networks," *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 4, Apr. 2006, pp. 2-12.

[Pus00] T. Pusateri, "Distance Vector Multicast Routing Protocol Applicability Statement", *IETF draft-ietf-idmr-dvmrp-v3-as-00.txt*, Jul. 2000.

[Ram97] R. Ramaswami, G. H. Sasaki, "Multi-wavelength optical networks with limited wavelength conversion," *Proceedings of IEEE INFOCOM'97*, Vol. 2, pp. 489-498.

[Ram98a] R. Ramaswami and K. N. Sivarajan, "Optical Networks: A Practical Perspective," *Morgan Kaufmann Publishers*, San Francisco, 1998.

[Ram98b] S. Ramamurthy and B. Mukherjee, "Fixed alternate routing and wavelength conversion in wavelength-routed optical networks," *Proceedings of IEEE Global Telecommunications Conference* (GLOBECOM'98), pp. 2295-2303.

[Ram02] S. Ramesh, G. N. Rouskas, and H.G. Perros, "Computing Call Blocking Probabilities in Multi-class Wavelength Routing Networks with Multicast Traffic," *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 1, Jan. 2002, pp. 89-96.

[Rou97] G. N. Rouskas, M. H. Ammar, "Multidestination communication over tunable-receiver single-hop WDM networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 3, 1997, pp. 501-511.

[Rou02] G. N. Rouskas, "Light-Tree Routing Under Optical Layer Power Budget Constraints," *Proceedings 17th IEEE Computer Communication Workshop*, Oct. 2002.

[Rou03] G. N. Rouskas, "Optical Layer Multicast: Rational, Building Blocks, and Challenges," *IEEE Network*, Vol. 17, No. 1, Jan./Feb. 2003, pp. 60-65.

[Sah98] G. Sahin, M. Azizoglu, "Multicast routing and wavelength assignment in wide area networks," *SPIE Conference on All-Optical Networking: Architecture, Control, and Management*, SPIE 3531 Vol. 3531, Nov. 1998, pp. 196-208.

[Sah99] L. H. Sahasrabuddhe and B. Mukherjee, "Light-trees: Optical Multicasting for improved Performance in Wavelength-routed Networks," *IEEE Communications Magazine*, Vol. 37, No. 2, Feb. 1999, pp. 67-73.

[Sai00] H. Saito, Y. Miyao, and M. Yoshida, "Traffic Engineering Using Multiple Multipoint-to-point LSPs," *Proceedings of IEEE INFOCOM'00*, Vol. 2, pp. 894-901.

[Spa00] J. Spath, "Dynamic routing and resource allocation in WDM transport networks," *Computer Networks,* Vol. 32, May 2000, pp. 519-538.

[Sre00] N. Sreenath, K. Satheesh, G. Mohan, C. Siva Ram Murthy, "Virtual source-based multicast routing in WDM optical networks," *Proceedings of IEEE International Conference on Networks (ICON)*, 2000, pp. 385-389.

[Sre01] N. Sreenath, N. Krishna Mohan Reddy, G. Mohan, C. Siva Ram Murthy, "Virtual source-based multicast routing in WDM networks with sparse light splitting" *Proceedings of IEEE workshop on High Performance Switching and Routing*, 2001, pp. 141-145.

[Sri03] R. Srinivasan and A. Somani, "Dynamic Routing in WDM Grooming Networks," *Photonic Network Communications*, Vol. 5, No. 2, 2003, pp. 123-135.

[Sub97] S. Subramania and R. Barry, "Wavelength assignment in fixed routing WDM networks," *Proceedings of IEEE International Conference on Communications (ICC)*, 1997, Vol. 1, pp. 406-410.

[Sub98] S. Subramaniam, M. Azizoglu, A. K. Somani, "On the optimal placement of wavelength converters in wavelength routed networks," *Proceedings of IEEE INFOCOM'98*, pp. 902-909.

[Tan96] A. S. Tanenbaum, "Computer Networks," Third Edition, *Prentice Hall Inc*, Upper Saddle River, New Jersey, 1996.

[Tse01] W. –Y. Tseng and S. –Y. Kuo, "All-Optical Multicasting on Wavelength-Routed WDM Networks with Partial Replication," *Proceedings of 15th International Conference on Information Networking*, pp. 813-818, 2001.

[Tse03] K. –B. Tseng and J. –F. Huang, "Optimal routing path and wavelength assignment in WDM multicast networks," *The 9th Asia-Pacific Conference on Communications, 2003*, (APCC 2003), Vol. 1, Sept. 2003, pp. 34 – 37.

[Var04] R. Varrazza, I. B. Djordjevic and S. Yu, "Active vertical coupler based optical crosspoint switch matrix for optical packet switching applications," *IEEE Journal of Lightwave Technology*, Vol. 22, No. 9, Sept. 2004, pp. 2034-2042.

[Var05] R. Varrazza, T. Bricheno and S. Yu, "A fully packaged 4×4 integrated optical switch matrix," *IEEE Journal of Selected Topics in Quantum Electronics*, Vol. 11, No. 6, Nov./Dec. 2005, pp.1248 – 1254.

[Wan96] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 7, Sept. 1996, pp. 1228-1234.

[Wan03] J. Wang, B. Chen and R. N. Uma, "Dynamic Wavelength assignment for multicast in all-optical WDM networks to maximize the network capacity," *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 8, Oct. 2003, pp. 1274-1284.

[Wan04] J. Wang, X. Qi, M. Yang and B. Chen, "Exploring core based tree (CBT) in WDM networks," *Proceedings of 7th International Symposium on Parallel Architectures, Algorithms and Networks*, May 2004, pp. 489 – 494.

[Wan05] X. Wang, H. Cheng, M. Huang and S.K. Das, "Flexible QoS multicast routing based on artificial immune algorithm in IP/DWDM optical Internet," *Proceedings of IEEE International Conference on Communications (ICC)*, 2005, Vol. 3, pp. 1631 – 1635.

[Wan06] Z. Wang, N. Chi, and S. Yu, "2-to-4 Optical Multicast Using Active Vertical Coupler Optical Crosspoint Switch Matrix," *IEEE Photonic Technology Letters*, Vol. 18, No. 1, Jan. 2006, pp. 286-288.

[Wan06a] J. Wang, Xi Qi, and B. Chen, "Wavelength Assignment for Multicast in All-Optical WDM networks with Splitting Constraints," *IEEE/ACM Transactions on Networking*, Vol. 14, No. 1, Feb. 2006, pp. 169-182.

[Wan06b] Z. Wang, N. Chi and S. Yu, "Characterization of 1×N broadcast and 2×N multicast packet switching using active-vertical-coupler-based optical crosspoint switch," *IEEE Journal of Lightwave Technology*, Vol. 24, No. 8, Aug. 2006, pp. 2978-2985.

[Wan06c] Z. Wang, N. Chi and S. Yu, "2-to-many lossless optical multicast using an optical crosspoint switch matrix," *Optical Fiber Communication Conference, 2006 and the 2006 National Fiber Optic Engineers Conference*, Mar. 2006, pp. 1-3.

[Wes96] D. B. West, "Introduction to Graph Theory", *Prentice Hall*, Upper Saddle River, NJ, 1996.

[Wu01] K.–D. Wu, J.C. Wu, and C.S. Yang, "Multicast Routing With Power Consideration in Sparse Splitting WDM Networks," *Proceedings of IEEE International Conference on Communications (ICC),* 2001, Vol. 2, pp. 513-517.

[Xin04] Y. Xin and G. N. Rouskas, "Multicast Routing under optical layer constraints,". *Proceedings of IEEE INFOCOM'04*, Vol. 4, pp. 2731-2742.

[Yan00] Y. Yang, J. Wang, and C. Qiao, "Nonblocking WDM multicast switching networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, Dec. 2000, pp. 1274-1287.

[Yan01] S. Yan, M. Ali and J. Deogun, "Route Optimization of Multicast Sessions in Sparse-Splitting Optical Networks," *Proceedings of IEEE Global Telecommunications Conference* (GLOBECOM'01), pp. 2134-2138.

[Yan02] S. Yan, "Multicast routing and Wavelength Assignment in Wavelength-Routed Optical Networks," *PhD's dissertation*, University of Nebraska, USA, 2002

[Yan03] D.-N. Yang and W. Liao, "Design of light-tree based logical topologies for multicast streams in wavelength routed optical networks," *Proceedings of IEEE INFOCOM'03*, Vol. 1, pp. 32 – 41.

[Yet03] E. Yetginer, E. Karasan, "Regenerator Placement and Traffic Engineering with Restoration in GMPLS networks," *Photonic Network Communications*, 6: 2, 2003, pp. 139-149.

[Yos06] S. Yoshima, K. Onohara, N. Wada, F. Kubota and K. Kitayama, "Multicast-capable optical code label switching and its experimental demonstration," *IEEE Journal of Lightwave Technology*, Vol. 24, No. 2, Feb. 2006, pp. 713-722.

[Yu05] O. Yu and Y. Cao, "Mathematical formulation of optical multicast with loss-balanced light-forest," *Proceedings of IEEE Global Telecommunications Conference* (GLOBECOM '05), pp. 1-5.

[Yu05a] S. Yu, S. Lee, O. Ansell and R. Varrazza, "Lossless optical packet multicast using active vertical coupler based optical crosspoint switch matrix," *IEEE Journal of Lightwave Technology*, Vol. 23, No. 10, Oct. 2005, pp. 2984-2992.

[Zan99] H. Zang, J. P. Jue, L. Sahasrabuddhe, R. Ramamurthy, B. Mukherjee, "Connection management for wavelength-routed WDM networks," *Proceedings of IEEE Global Telecommunications Conference* (GLOBECOM'99), Vol. 2, pp. 1428-1432.

[Zan00] H. Zang, J. P. Jue, and B. Mukherjee, "A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks," *Optical Networks Magazine*, Vol. 1, No. 1, Jan. 2000, pp. 47-60.

[Zan01] H. Zang, J. P. Jue, L. Sahasrabuddhe, R. Ramamurthy, B. Mukherjee, "Dynamic Lightpath Establishment in Wavelength-Routed WDM networks," *IEEE Communications Magazine*, Vol. 39, No. 9, Sept. 2001, pp. 100-108.

[Zha99] X. Zhang, "Traffic grooming & scheduling, Wavelength routing & assignment, and multicast in WDM networks," *PhD's dissertation*, State University of New York at Buffalo, USA, 1999.

[Zha00] X. Zhang, J. Y. Wei, and C. Qiao, "Constrained Multicast Routing in WDM Networks with Sparse Light Splitting," *IEEE Journal of Lightwave Technology*, Vol. 18, No. 12, Dec. 2000, pp. 1917-1927.

[Zho04] Y. Zhou and G. S. Poo, "A new multiwavelength multicast wavelength assignment (MMWA) algorithm in wavelength-routed WDM networks," *Proceedings of IEEE International Conference on Communications (ICC),* 2004, Vol. 3, pp. 1786 – 1790.

Appendix A Confidence Interval Calculation

The accuracy of simulation data estimate can be indicated by Confidence Intervals (CI). For an unknown population parameter X, the Confidence Interval provides an estimated range of values. Given a small probability α, the Confidence Interval means there is only a probability of α that the true expectation lies outside that range.

The intervals can be calculated from a set of sample data. Suppose we have a succession of N independent simulations runs. Each of them is long enough so that the results are not correlated. The mean results, e.g., blocking probabilities, obtained from these runs are specified as X_1, X_2,..., X_i,..., X_N. Student t-test can be used to calculate the Confidence Interval from this set of data. The range of the Confidence Interval is denoted by $[L, U]$, where L means the lower bound for the population and U represents the corresponding upper bound. Using the standard deviation and the t distribution table, the lower bound and the upper bound can be calculated as follows:

$$L = \overline{X} - \frac{S_X t_{[1-\frac{\alpha}{2}, N-1]}}{\sqrt{N}}$$

$$U = \overline{X} + \frac{S_X t_{[1-\frac{\alpha}{2}, N-1]}}{\sqrt{N}}$$

Where:

$$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N}$$ represents the sample mean of all N runs;

$$S_X = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \overline{X})^2}{N-1}}$$ is the sample standard deviation;

and $t_{[1-\frac{\alpha}{2}, N-1]}$ is the $1-\frac{\alpha}{2}$ quantile of the student t-distribution with N-1 degree of freedom.

Calculated from a given α, the interval of $[L, U]$ is called a 100(1-α)% confidence interval. It means that with 100(1-α)% probability, the true value of the parameter will be included in the interval.

We calculated 90% confidence intervals for our simulation data points, which means α is 0.10. Sample values are obtained from 5 independent runs. From the table of the t distribution, $t_{[1-\frac{\alpha}{2}, N-1]}$ is found to be 2.132.

The following table provides an example calculation of blocking probability confidence intervals. The parameters correspond to the simulation environment where *S = 50%, M=3, T=50%, W=16.*

Table A-1: Example of CI calculation

Load	*Blocking Probability average in each run*					$\overline{X}$	S_X	*L*	*U*
	X_1	X_2	X_3	X_4	X_5				
10.2	0.9346	0.9349	0.9492	0.9512	0.9529	0.9446	9E-3	0.9360	0.9532
9.6	7E-4	1.7E-4	2.1E-4	1.1E-4	3.3E-4	3.0E-4	2.3E-4	0.8E-4	5.2E-4