A study of routing and wavelength assignment problems in wavelength routed optical networks

Wang, Ying

2008

Wang, Y. (2008). A study of routing and wavelength assignment problems in wavelength routed optical networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41416>

<https://doi.org/10.32657/10356/41416>

A Study of Routing and Wavelength Assignment Problems in Wavelength Routed Optical Networks

Wang Ying

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2008

Statement of Originality

I hereby certify that the content of this dissertation is the results of work done by me and has not been submitted for a higher degree to any other university or institution.

15 April 08

Date

Wang Ying

Acknowledgments

I would like to express my deepest appreciation to my supervisor, Assoc. Prof. Cheng Tee Hiang, who has provided me with invaluable guidance and advice whenever I needed them. His unwavering support and encouragement have made my research experience a rewarding and pleasant one.

I would like to extend my heartfelt appreciation to two other professors of the School of Electrical & Electronics Engineering; namely, Assoc. Prof. Sanjay Kumar Bose and Assoc. Prof. Lim Meng Hiot. They are very generous in sharing their wealth of technical knowledge and insights.

I am also grateful to my friend Shen Gangxiang of the University of Alberta for the useful advice and suggestions he gave.

I am thankful to the staff members of Network Technology Research Centre (NTRC) and Institute for Infocomm Research (I²R), in particular, Assoc. Prof. Lu Chao, Assoc. Prof. Shum Ping and Miss Nancy Ooi for their kind assistance. I am grateful to the postgraduate students in my laboratory for all the interesting discussions and wonderful time we had.

I am thankful to Nanyang Technological University for awarding me the research scholarship.

Finally, I wish to dedicate this thesis to my family, who has given me unconditional love and care.

Summary

Devising a good *routing and wavelength assignment* (RWA) scheme is a very important and challenging problem to be addressed in the design of a *wavelength routed optical network* (WRON). RWA deals with finding a route from a network of physical links for each given pair of source and destination nodes, and appropriately selects and reserves a specific unused wavelength on each of the links along the route. In the context of WRON networks, RWA related problems could be classified into four categories in accordance with the type of traffic pattern and the stage of the design and planning process; they are, namely, *static lightpath establishment* (SLE) problem, *dynamic lightpath establishment* (DLE) problem, *virtual topology design* (VTD) problem, and *dynamic label switched path provisioning* (DLP) problem.

In this dissertation, we describe and summarize various representative solution techniques for the above-mentioned problems. For some specific problems, we manage to find better solution techniques that outperform the best known solutions in one or more ways.

For the SLE problem, solutions proposed in past studies are considered computationally expensive, especially when large networks are concerned. We experiment with soft-computing techniques and have some success in using *tabu search* (TS) to solve the SLE problem. The simulation results obtained confirm that our proposed TS algorithm has excellent performance for both small and large networks. In addition, we propose a new way to formulate the SLE problem so that it can be used to optimize the revenue for existing network resource (*maximum revenue*

problem). To our best knowledge, we are the first to formulate the SLE problem that way to make it useful for network operators. We also provide solutions based on *integer linear programming* (ILP) and TS for this problem.

For the DLP problem, network control can be classified as falling under the overlay, augmented, or peer model. For the first two models, the IP layer and optical layer only have the resource information of their respective layers. For the peer model, all the resource information is known to both layers; hence, it can potentially achieve better resource utilization. Unfortunately the peer model approach is generally not acceptable to the operators if the IP layer and optical layer belong to two different administrations. In addition, it is much more complex and generally seen as not feasible in the near future. For this reason, we devise an enquiry mechanism that enables the IP layer to obtain the resource information of the optical layer for the overlay model. The enquiry mechanism is simple enough such that it does not significantly increase the complexity of the *labeled-switched path* (LSP) provisioning process. Two simple and novel algorithms for LSP requests based on the proposed enquiry mechanism are proposed. Simulation results show that the proposed algorithms perform significantly better than other routing algorithms for the overlay model and are nearly as good as the integrated routing algorithm for the peer model.

The aforementioned TS algorithm for the SLE problems, new SLE problem formulation and solutions for optimizing revenue, the enquiry mechanism and the novel algorithms for handling LSP requests for the overlay model for the DLP problem, and the detailed performance studies comparing the proposed algorithms and other algorithms form the bulk of the original contributions in this dissertation.

Contents

Summary 3

List of Figures 8

Chapter 1 Introduction 11

1.1 Motivation..... 13

1.2 Objective 17

1.3 Scope and Contribution of the Dissertation 18

1.4 Organization of the Dissertation 20

Chapter 2 Wavelength Routed WDM Optical Networks and IP-over-Optical Networks 22

2.1 Wavelength Routed WDM Optical Networks 22

2.2 IP-over-Optical Networks 25

Chapter 3 Literature Review of Related Works 31

3.1 Static Lightpath Establishment (SLE) Problem 32

3.1.1 Integer Linear Programming Solutions..... 32

3.1.2 Heuristic Solutions..... 36

3.2 Dynamic Lightpath Establishment (DLE) Problem..... 39

3.2.1 Routing..... 42

3.2.2 Wavelength Assignment 47

3.3 Virtual Topology Design (VTD) Problem 48

3.3.1 ILP Formulations 49

3.3.2 Heuristic Algorithms..... 51

3.4 Dynamic LSP Provisioning (DLP) Problem..... 52

3.5	Other Relevant Research Work	55
Chapter 4 Tabu Search Solutions for the SLE Problems		58
4.1	Generic Tabu Search Algorithm	58
4.2	MLE Problem with TS Solution	64
4.2.1	Trial Move Selection	67
4.2.2	Tabu List	68
4.2.3	Aspiration Criteria	69
4.2.4	Stopping Condition	69
4.2.5	Pseudo Codes for the Proposed TS Algorithm	70
4.2.6	Simulations	73
4.3	MWU Problem with TS Solution	80
4.3.1	Heuristic Algorithm Using MLE TS Algorithm	80
4.3.2	Simulations	81
4.4	Conclusions and discussion	83
Chapter 5 Maximum Revenue Problem and Solutions.....		85
5.1	ILP Formulation.....	86
5.2	Heuristic Algorithms.....	89
5.2.1	TS Algorithm	89
5.2.2	Sequential RWA Algorithm.....	94
5.3	Simulations	95
5.3.1	Performance Comparison for Small Networks	95
5.3.2	Performance Comparison for a Large Network.....	100
5.4	Conclusions and discussions.....	102
Chapter 6 Enquiry Process for IP-over-optical Networks Based on the Overlay Model.....		104

6.1	Proposed Enquiry Mechanism	107
6.1.1	Enquiry Process	110
6.1.2	Enquiry Message Format	113
6.1.3	Resource Allocation.....	117
6.3	Provisioning Scheme with Enquiry Process	117
6.4	Conclusions.....	118
Chapter 7 Routing Algorithms for Enquiry-based Provisioning Scheme		120
7.1	Proposed Routing Algorithms.....	123
7.1.1	Integrated Routing with Estimated Link Cost (IRELC)	126
7.1.2	Integrated Routing with Tree Enquiry (IRTE)	128
7.2	Simulations	130
7.2.1	Effect of Bandwidth Granularities of LSP Requests	131
7.2.2	Performance Comparison for Dynamically Varying Bandwidth Requirement.....	137
7.3	Conclusions and discussions.....	140
Chapter 8 Conclusions and Recommendations		141
8.1	Conclusions.....	141
8.2	Suggestions for Further Research	143
List of Author's Publications		145
References		146
Acronyms		151

List of Figures

Figure 1.1	A wavelength routed optical network	12
Figure 2.1	A core WRON network.....	22
Figure 2.2	OXC structure	23
Figure 2.3	An IP-over-optical network model.....	28
Figure 2.4	Example of a simple control plane node architecture	29
Figure 3.1	Graph conversion in FAA	41
Figure 3.2	Routing topology in IRA.....	54
Figure 4.1	Test topologies	74
Figure 4.2	Result for 10 trials with $ \Delta =500$	82
Figure 5.1	Results for NSFNet.....	97
Figure 5.2	Results for ARPANet.....	98
Figure 5.3	Results for BigNet.....	101
Figure 5.4	Running time comparison	102
Figure 6.1	Routing comparison between overlay model and peer model	105

Figure 6.2	Example of IP-over-Optical network	109
Figure 6.3	Enquiry process scheme A	111
Figure 6.4	Enquiry process scheme B	112
Figure 7.1	Example for an enquiry-based routing algorithm	120
Figure 7.2	Operation of the proposed algorithms	123
Figure 7.3	Effect of bandwidth granularity of LSP requests	134
Figure 7.4	Performance results for different granularities	136
Figure 7.5	Performance comparison for different dynamic bandwidth granularities	138
Figure 7.6	Performance comparison for different static and dynamic bandwidth granularities	139

List of Tables

Table 4.1 Tabu tenure effects for NSFNet 76

Table 4.2 Tabu tenure effects with BigNet 76

Table 4.3 Number of accepted requests for ten executions..... 77

Table 4.4 Statistics 77

Table 4.5 ILP and TS results for BigNet..... 79

Table 5.1 Results for NSFNet 99

Table 5.2 Results for ARPANet..... 100

Chapter 1

Introduction

The thirst for bandwidth fuelled by an explosive increase in the Internet traffic makes *wavelength division multiplexing* (WDM) the most promising technology for implementing core optical transport networks and *metropolitan area networks* (MANs)[1]. In WDM, the optical transmission spectrum is divided into a number of non-overlapping wavelength (or frequency) bands, with each wavelength supporting a single communication channel operating at its peak electronic speed. The recent announcement of a 160 x 10Gb/s WDM system demonstrates the potential of WDM in meeting future explosive demand of bandwidth [2].

In future, WDM will not only be used in point-to-point transmission systems where a wavelength will be terminated at each network node but also in a wavelength routed network where end-to-end optical channel trails can be set up to link different pieces of subscriber premise equipment [3]. This is illustrated in Figure 1.1. In such a wavelength routed optical network, there are many *optical cross-connects* (OXCs) interconnected by bidirectional fibers to form a mesh topology. Edge nodes act as interfaces between the optical core and non-optical subscriber premise equipment – such as IP routers, *asynchronous transfer mode* (ATM) switches, etc. Access points perform optical-to-electronic and electronic-to-optical conversions, and are the end

points for the optical paths. The communication paths between different pieces of end user equipment may continue outside the optical core of the network in electrical form.

Figure 1.1 A wavelength routed optical network

The services that a wavelength routed network offers to its higher layer clients are in the form of logical connections implemented using *lightpaths*. A lightpath, which is an optical channel trail traversing single or multiple fiber spans, can be set up from a source access (ingress) node to a destination access (egress) node [4]. Traffic transmitted on a lightpath does not need to undergo any electronic conversion within the optical network if electronic wavelength conversion is not performed along the lightpath. In other words, a lightpath can be “transparent” to the format and encoding

of higher layer traffic; hence, can be used to carry any kind of service. Since a lightpath is regarded as a logical connection, flexible routing strategies can be implemented such that the network can adapt to any traffic changes in the higher layer or to recover from any failure in the network.

1.1 Motivation

There is a fundamental difference between a *wavelength routed optical network* (WRON) and a traditional circuit-switched network in that a connection in the former is associated with two attributes, namely, the route and the chosen wavelengths while the connection in the later only has one attribute; i.e., the route. A lightpath is implemented in WRON by selecting a route of physical links between two end nodes, and selects and reserves a particular wavelength for each of those links that constitute the route. The problem of selecting a route and assigning a wavelength to that route based on certain design objectives and constraints is generally dubbed the *routing and wavelength assignment* (RWA) problem [5].

The traffic patterns of WRON can be classified as *static* or *dynamic*. For the case of static traffic, all the connection requests are assumed to be known in advance, the goal of the RWA problem is to set up lightpaths for all the connection requests so that network resources such as the number of wavelengths and fibers in the network can be minimized. Another widely studied problem for static traffic is to maximize the number of connection requests that can be fulfilled with a given set of connection requests. The RWA problems associated with static traffic are commonly known as *static lightpath establishment* (SLE) problems [5].

The study of SLE problem is an important step in network planning. In planning for a new WRON or in reviewing the design of an existing WRON for upgrading, an operator needs to forecast the traffic demand (lightpath requests), evaluates various network architecture alternatives, and considers the technical specifications and cost of various systems and subsystems to be used. Also, the operator needs to decide on the physical locations of OXCs, the physical layout of transmission facilities, and how to realize existing and anticipated new lightpath requests such that the capital expenditure (capex), and in some cases operational expenditure (opex) also, can be minimized.

When a network operator needs to plan for the migration of a legacy optical network, say *synchronous digital hierarchy* (SDH) network, to a WRON, the most logical and pragmatic approach is to make use of the existing network facilities as much as possible. This more or less dictates the locations of network nodes and the physical links between nodes; which can be used as input parameters for network planning. From the research point of view, since it is hard to construct a cost model for network equipment and transmission infrastructure, instead of directly attacking the cost optimization problem, most previous research studies focus on minimizing the number of wavelengths that need to be used in a fiber link. This design objective is motivated by the observation that the complexity of an OXC increases with the number of wavelengths it has to switch; hence, and the cost of the transmission facilities will increase when the number of wavelengths per fiber increases. In other words, a smaller number of wavelengths used implies lower cost though the cost cannot be easily quantified and there is no guarantee that least number of wavelengths used does lead to minimum cost.

For the case of dynamic traffic, connection requests are assumed to reach the network in a random manner. For each connection request that arrives, the operator needs to set up a lightpath based on the existing network status. If there are not enough resources to accommodate the request, the request is rejected by the network. Each lightpath established in the network will be torn down when the client initiates a termination request; the corresponding resource will be released when the process is completed. The RWA problem for dynamic traffic is commonly known as the *dynamic lightpath establishment* (DLE) problem [5]. The DLE problem arises when an operator offers dynamic switched lightpath services. In this case, the operator does not know when its customers will initiate a new lightpath request and terminate an existing lightpath. When a new lightpath request is initiated by a customer, the operator needs to consider how to set up the lightpath on-the-fly based on the network state, its operation procedure and other considerations.

Recently the popularity of Internet has led to an explosive increase in the IP data traffic, which overwhelms the traditional voice traffic. This has a major impact on the architecture and design of existing and future transport networks. The most-talked-about network architecture is IP-over-optical network [4][6][7]. In this structure, the IP layer is built directly on top of the optical layer, and the lightpath connections in the optical layer define the logical links between the IP network nodes.

Logically, RWA related problems for IP-over-optical networks can be categorized according to whether the IP traffic is static or dynamic. By static IP traffic we mean the IP traffic flow between different pairs of source and destination IP nodes is characterized mainly by a set of static values that quantify the amount of bandwidth required (which can be based on the average or peak IP traffic flow or some other

parameters) between these node pairs. The RWA related problems in this case are classic *virtual topology design* (VTD) problems [8][9]. During the planning stage in deploying an IP network over a WRON, the operator needs to decide where to place the logical links between IP layer network nodes to form the IP layer topology, and normally it involves meeting certain design objectives and constraints. In this case, the operator would have the forecasted IP traffic demand in term of bandwidth required, which can then be considered as static traffic for planning purpose. Since a logical link in the IP layer is essentially a lightpath connection in the optical layer, the design of a logical topology that is optimal or near optimal based on certain design objectives need to simultaneously consider the IP and optical layer resources; in particular, the routes and wavelengths used in the optical layer.

For an up and running IP network, the IP traffic as seen by the operator for network design and planning purpose is dynamic. With the wide-spread adoption of *multi-protocol label switching* (MPLS)[10] protocol in the IP network, IP traffic will essentially be transmitted via a virtual-circuit-like *label switched path* (LSP) from an ingress *label switching router* (LSR) to an egress LSR. With the introduction of the *generalized multi-protocol label switching* (GMPLS)[11] signaling framework, a lightpath can be regarded as a higher order LSP, which defines the logical topology of the IP layer and provides spare capacity for IP traffic. With GMPLS, the IP traffic demand is now regarded as an LSP request with certain bandwidth requirement.

LSP requests can be routed solely in the IP layer; however, from resource utilization point of view, routing an LSP request while considering the spare resource in both IP and optical layer can achieve better network performance. Two-layer routing will involve the process of finding lightpaths in the optical layer as well as

LSPs in the IP layer; thus, it is related to RWA problems. In the following, we dub the problems related to dynamically provisioning LSP requests in the two-layer fashion as *dynamic LSP provisioning* (DLP) problems.

1.2 Objective

Depending on the traffic assumptions and how a wavelength routed optical network (WRON) will evolve from a legacy SDH network and integrated with the IP layer, network architects are faced with different types of routing and wavelength assignment (RWA) design problems. These problems have been studied quite extensively but by no means complete. The objectives of our study are two folds: firstly, to identify and solve niche but important RWA problems in WRONs; and secondly, where applicable, devise better solution techniques that will outperform the best known solutions in one or more ways.

In fulfilling the aforementioned objectives, we identify the following three specific RWA problems as the foci of our studies:

1. Devise a more computationally efficient approach for solving the static lightpath establishment (SLE) problem.
2. Devise a model and an optimization technique that would allow network operators to maximize the revenue of a WRON.
3. Devise a computationally efficient integrated routing scheme for an overlay IP-over-optical network

1.3 Scope and Contribution of the Dissertation

The studies of RWA problems in optical networks involved solving different sub-problems belonging to the SLE, DLE, VTD, and DLP problems. In this dissertation, we will review different representative solution techniques for the above-mentioned problems based on an extensive review of literatures in the public domain. The solution techniques reviewed may not be the best solution to the problem studied; they are selected because they make use of some unique methodologies. For some specific problems, we have managed to find better solution techniques that outperform the best known solutions in one or more ways.

The contribution of this dissertation can be summarized as follow:

4. Though SLE, DLE, VTD, and DLP routing and wavelength assignment problems have been widely studied individually, there is no cohesive study of all the aforementioned problem domains in a collective manner. The study reported in this thesis aims to fill that void.
5. For the SLE problem, solutions provided by past research are generally regarded as computationally expensive, especially when large networks are concerned. Another contribution is related to our success in using *tabu search* (TS) to SLE problem. In this work, the performance of the proposed TS algorithm is compared with the *integer linear programming* (ILP) solution, which sets the upper bound for optimality. The results for both small and large networks show that the proposed TS algorithm works

almost as well as the ILP solution and is much more computationally efficient.

6. The SLE problem is usually formulated as an optimization problem with the objective of minimizing wavelength channel usage or maximizing the number of connections established. Another contribution is related to a new problem formulation that aims to maximize network revenue, which is more appealing to network operators compared to other problem formulations that aim to minimize the number of wavelength per fibre link. To our best knowledge, we are the first to develop the model, and solutions based on ILP formulation and TS.
7. For the DLP problem, network control can be classified as falling under the overlay, augmented, or peer model. For the first two models, the IP layer and the optical layer only have access to the resource information of their respective layers. For the peer model, all the resource information is known to both layers; hence, it can potentially achieve better resource utilization. Unfortunately, the peer model approach is generally not acceptable to the operators if the IP layer and optical layer belong to two different administrations. In addition, it is much more complex and generally seen as not feasible in the near future. For this reason, we propose an enquiry mechanism, which enables the IP layer to obtain the resource information of optical layer for the overlay model. The enquiry mechanism is efficient in collecting optical layer resource information and simple enough such that it does not significantly increase the complexity of the *labeled switched path* (LSP) provisioning process.

8. Based on the enquiry mechanism, two simple and novel algorithms for routing LSP requests are proposed. Simulation results show that our proposed algorithms perform significantly better than other routing algorithms for the overlay model and are nearly as good as the integrated routing algorithm for the peer model.

1.4 Organization of the Dissertation

The reminder of this dissertation is organized as follows:

We devote Chapter 2 to the introduction of concepts and background information relevant to this dissertation. A general introduction to WRON networks is given and issues related to IP-over-optical network structure are discussed.

In Chapter 3, important sub-problems in the SLE, DLE, VTD and DLP problem domains are discussed and solution techniques developed in past studies are reviewed.

In Chapter 4, our tabu search algorithms for the SLE problem are described, and simulation results are presented and analyzed.

In Chapter 5, the problem of maximizing revenue subject to certain constraints is studied, followed by a description of the ILP solution and TS algorithm and analysis of results.

Chapter 6 describes our proposed enquiry mechanism.

In chapter 7, two simple and novel routing algorithms to be used in conjunction with the enquiry mechanism are presented.

In Chapter 8, we conclude the work described in the dissertation and discusses possible directions in which the work can be extended.

Chapter 2

Wavelength Routed WDM Optical Networks and IP-over-Optical Networks

2.1 Wavelength Routed WDM Optical Networks

A wavelength routed WDM optical network is shown in Figure 2.1. It consists of a number of *optical crossconnects* (OXCs) interconnected by bidirectional optical links. Typically each bidirectional link consists of a pair of unidirectional optical fibers with optical signals in the two fibers propagating in the opposition direction.

Figure 2.2 OXC structure

An OXC is basically a wavelength (or lambda) router, which is capable of switching individual wavelengths on an incoming link to any outgoing link. If every link has the same wavelength of W , an OXC can be regarded as a combination of W independent space-switching matrices (Figure 2.2). Each space-switching matrix has F input ports and F output ports, where F is the number of incoming/outgoing links of the OXC (We hereby assume that the OXC has the same number of incoming and outgoing links). The wavelengths multiplexed on the incoming links will be de-

multiplexed and fed to the corresponding switching matrices. Add and drop ports need to be present in those OXC's which double-up as sources and sinks for the traffic of higher layer clients. Normally an add port consists of an electronic-to-optical converter, which is the transmitter or source of a wavelength channel, while a drop port consists of an optical-to-electronic receiver, which is the receiver or sink of a wavelength channel.

Depending on whether an OXC has the ability to convert a channel from one wavelength to another, an OXC can be classified as a *wavelength-interchanging crossconnect* (WIXC) or a *wavelength-selective crossconnect* (WSXC). WIXC has the ability to change the wavelength of an incoming optical channel to another wavelength of an outgoing optical channel while WSXC simply switches the optical signal from an incoming fiber to an output fiber without changing the wavelength [12]. Existing WIXCs are O-E-O crossconnects that switch data channels by first converting the optical signals into electronic signals while WSXCs are O-O-O crossconnects that do not convert optical signals into electronic signals. Though the benefits of wavelength conversion are well known [13][14], the high cost of optical wavelength converters and inefficiency of O-E-O wavelength converters [15] will hinder their widespread adoption in optical networks; thus, it is more realistic to consider an optical network with only WSXCs.

A lightpath is defined as an end-to-end optical channel trail between an ingress node and an egress node in the network [4]. If there is no WIXC in the network, a lightpath needs to use the same wavelength on all the links along the route. In other words, *wavelength continuity constraint* (WCC) has to be strictly followed for the lack of wavelength conversion. A lightpath can be represented as a duplet (p, w) ,

where p is the route and w is the wavelength assigned to the lightpath. After a lightpath is established, the resources allocated are used exclusively during the holding time of that lightpath. For this reason, two lightpaths which exist in the network at the same time cannot use the same wavelength on any link which they share. We dub this constraint the *lightpath conflict constraint* (LCC). Examples of lightpaths that adhere to the WCC and LCC constraints are exemplified in Figure 2.1. In Figure 2.1, there are two connection requests in progress, one from OXC_1 to OXC_3 , and the other from OXC_1 to OXC_4 . The first request is fulfilled by lightpath L_1 . L_1 is routed along OXC_1 , OXC_2 and OXC_3 . To follow the WCC, the same wavelength λ_1 is assigned to L_1 on links $OXC_1 \rightarrow OXC_2$ and $OXC_2 \rightarrow OXC_3$. The second request is fulfilled by lightpath L_2 , which is routed along OXC_1 , OXC_2 and OXC_4 and assigned wavelength λ_2 to avoid conflict with L_1 on link $OXC_1 \rightarrow OXC_2$.

2.2 IP-over-Optical Networks

Recently the popularity of Internet has led to an explosive increase in IP data traffic, which overwhelms the traditional voice traffic; obviously this has a major impact on the design of existing and future transport networks. Today, a typical telecommunication network has four layers: IP and other content-bearing traffic at the top layer, an *asynchronous transfer mode* (ATM) network functioning as the service aggregation point at the second layer, a *synchronous digital hierarchy* (SDH) network functioning as a transport network at the third layer, and a *wavelength division multiplexing* (WDM) network at the bottom layer. This architecture is favoured by incumbent operators in the near term; however, the well-known inefficiency of

maintaining a complicated multiplayer network will eventually force these operators to seek a more cost effective network structure. A much-talked-about solution is IP-over-optical network. In this network architecture, the redundant network layers are eliminated by emigrating some of the functions of ATM and SDH layers to IP and optical layer.

To bring the IP and optical layer together, new signaling and routing mechanism is needed for the control plane. For this reason, a framework of *generalized multi-protocol label switching* (GMPLS)[11] has been developed by the *Internet Engineering Task Force* (IETF). The GMPLS concept is originated from the existing *multi-protocol label switching* (MPLS)[10] framework. In MPLS, *forwarding equivalent class* (FEC) is used to identify a distinct flow of IP packets, which need to be forwarded over the same path and treated in the same manner by the network. For each FEC, a *label switched path* (LSP), which is a virtual circuit in the MPLS domain and represented as a sequence of *label switching routers* (LSR), is established via a *label distribution protocol* (LDP)[16] and *resource reservation protocol* (RSVP)[17]. For each IP packet that enters the MPLS domain, the ingress LSR attaches an MPLS label as the prefix to the packet. The label identifies the FEC to which the packet belongs and enables the packet to be label-switched by all the intermediate LSRs along the LSP to egress LSR. GMPLS generalizes the FEC and label concepts of MPLS to cover other types of resource and facility in an IP-over-optical network, which include packets for a *packet-switching capable* LSR (PSC-LSR), time-slots for a *TDM-switching capable* LSR (TSC-LSR, e.g., SDH-DXC), wavelengths for a *lambda-switching capable* LSR (LSC-LSR, e.g., OXC), and fibers for a *fiber-switching capable* LSR (FSC_LSR)[18]. With the generalized label concept, MPLS

protocol can be used in the optical domain, which makes fast automatic lightpath provisioning possible. In this dissertation, we only consider an IP-over-optical network with PSC-LSRs and LSC-LSRs because the other two types of LSRs are not relevant. For simplicity, we will use LSR to denote PSC-LSR and OXC for LSC-LSR in the rest of this dissertation.

The label stacking mechanism defined in GMPLS allows lower order (low granularity) LSPs to be multiplexed into an existing high-order LSP, which acts as a virtual link for the lower order LSPs; this way, low granularity traffic can be easily aggregated. Since a lightpath is basically a high-order LSP in the IP layer, we will use the term λ -LSP to refer to a lightpath in the optical layer.

In an IP-over-optical network, there exists a data plane that is responsible for data traffic forwarding and a control plane that is responsible for network control. Examples of network control functions include routing and wavelength assignment, protection and restoration, connection establishment and maintenance, and network state monitoring. The control plane is separated from the data plane because in the optical layer control messages are exchanged between different network elements via control channels, which are physically separated from the data channels. Such separation is required so that any failure in the control plane will not affect data transmission and any failure in the data plane will not jeopardize network control [19].

An OXC is a node in the control plane as well as in a data plane. Its data plane function is to switch wavelengths from input ports to output ports and its control function is to collaborate with other OXCs and other network control entities to exchange network control information for various optical network control functions

such as routing and wavelength assignment, and setting up and maintenance of lightpaths. Similarly, an LSR is also a node in the data plane and the control plane. Its data plane function is to switch date packets based on labels and its control plane function is to collaborate with other LSRs and network control entities for routing, LSP establishment and maintenance, and congestion control etc.

Figure 2.3 An IP-over-optical network model

An IP-over-optical network is depicted in Figure 2.3. As shown in the figure, high performance IP/MPLS routers are interconnected by WDM optical core networks. The optical layer is assumed to be incapable of processing individual IP packets and can only provide connectivity in the form of fixed-bandwidth lightpaths to the IP layer. The collection of lightpaths created in the optical layer then defines the logical topology of the IP/MPLS router network. The control planes of both IP and optical

layers are linked via *user-to-network interfaces* (UNIs). Through a UNI, the IP layer can signal the optical layer to set up a new lightpath or tear down an existing lightpath. Various optical sub-networks are connected via *network-to-network interfaces* (NNI), which allow routing, signaling and other control information to be exchanged between different optical sub-networks.

The GMPLS control plane is responsible for maintaining network status information and managing the creation, tearing down, and maintenance of connections. These functions may be implemented in different modules within each control plane node, as shown in Figure 2.4.

Figure 2.4 Example of a simple control plane node architecture

The routing module is responsible for advertising the availability of network resource (e.g. bandwidth on each wavelength) and other network attributes and constraints. An IP link state routing protocol such as OSPF [20] or IS-IS [21] with the necessary enhancement and extension is implemented in this module.

The link management module is responsible for addressing link management issues. This module deals with neighbor discoveries and link-state maintenance; for example, keeping track of up/down status of individual links. Typically, *link management protocol* (LMP) [22] developed by IETF will be implemented in this module.

The resource management module manages the local crossconnect resource and implements traffic engineering. The information disseminated by routing protocol is used by this module to form a local representation of network topology.

The signaling module is responsible for the signaling process associated with connection provisioning, deletion and failure recovery. For MPLS to be used in the optical network, such MPLS enhancements as RSVP-TE [23] and CR-LDP [24] need to be implemented.

Chapter 3

Literature Review of Related Works

As mentioned in Section 1.1 of this dissertation, for WRON networks, RWA related problems can be classified into the following four categories:

1. *Static lightpath establishment* (SLE) problem: A set of lightpath requests are given, and as many of these requests need to be fulfilled as possible, subject to a number of constraints. This problem can be solved in an off-line manner.
2. *Dynamic lightpath establishment* (DLE) problem: In this problem, lightpath request arrivals are not known in advance but in a random manner. Upon the arrival of a request, the lightpath needs to be provisioned in an online (real-time) manner.
3. *Virtual topology design* (VTD) problem: For this problem, the bandwidth requirement of each IP node pair is given. Lightpaths in the optical layer need to be determined off-line in a globally optimized fashion to form the IP layer topology.
4. *Dynamic LSP provisioning* (DLP) problem: For this problem, LSP requests with certain bandwidth requirement arrive dynamically. For each LSP request, lightpaths in the optical layer may be set up to accommodate the request in an on-line manner.

In what follows, we review different representative solution techniques for the above-mentioned problems. The solution techniques we review are not necessarily the best solution to the problem studied; they are reviewed because of some uniqueness in their methodologies. For some specific problems, we have managed to find better solution techniques that outperform the best known solutions in one or more ways; these new solution techniques will be described in later chapters.

3.1 Static Lightpath Establishment (SLE) Problem

For the SLE problem, lightpath requests are assumed to be known in advance and the RWA operation is to be carried out in an off-line manner. A commonly studied objective of SLE is to minimize the number of wavelengths in the network and this class of SLE problem is known as the *minimum wavelength usage* (MWU) problem [25][26][27]. For the MWU problem, a set of connection requests are given; the goal is to set up lightpaths for all the connection requests such that the number of wavelengths used is minimized. Another widely used objective of SLE is to maximize the number of connection requests that can be fulfilled by the network for a given set of wavelengths [28][29]. This is known as the *maximum lightpath establishment* (MLE) problem.

3.1.1 Integer Linear Programming Solutions

SLE problems are essentially optimization problems and can be formulated as *integer linear programming* (ILP) problems [25][26][27][28]. ILP is a branch of *mathematical programming*, which is a mathematical technique concerned with finding the best solution to a problem. The building of an ILP for an optimization

problem involves abstracting its essence and representing it as a series of mathematical relationships. Based on different abstractions of the SLE problems, various ILP formulations are possible and can be classified into link-based and path-based formulations [29]. In a link-based formulation, a lightpath request is treated as a flow in the network, and mathematic equations that describe flow balance at each network node for the constraint equations. An example of a link-based ILP formulation [29] for the MLE problem is given below. We have rewritten the formulation to make it consistent with other formulations presented in this dissertation.

Link-based ILP formulation (LILP):

Objective:

$$\text{Maximize } \sum_{s,d,w} L_{sd}^w \quad (3-1)$$

Subject to the following constraints:

$$\sum_i F_{sd,w}^{ij} - \sum_k F_{sd,w}^{jk} = \begin{cases} -L_{sd}^w & \text{if } s = j \\ L_{sd}^w & \text{if } d = j \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in N, (s,d) \in D, w \in W \quad (3-2)$$

$$\sum_w L_{sd}^w \leq \lambda_{sd}, \forall (s,d) \in D \quad (3-3)$$

$$\sum_{s,d} F_{sd,w}^{ij} \leq 1, \forall (i,j) \in E, w \in W \quad (3-4)$$

$$L_{sd}^w \in \{0,1\} \quad (3-5)$$

$$F_{sd,w}^{ij} \in \{0,1\} \quad (3-6)$$

Here, N denotes the set of nodes and E denotes the set of unidirectional links in the given network, and W is the set of wavelength channels available on each link. λ_{sd} denotes the number of requests in the given lightpath request matrix for source node s and destination node d . We use D to notate the set of all node pairs that have non-zero requests.

The binary decision variable L_{sd}^w is used to represent a lightpath: $L_{sd}^w = 1$ if there exists a lightpath from s to d on wavelength w ; otherwise, $L_{sd}^w = 0$. The binary decision variable $F_{sd,w}^{ij}$ is used to represent the flow of the lightpath L_{sd}^w : $F_{sd,w}^{ij} = 1$ if lightpath L_{sd}^w traverses link $i \rightarrow j$; otherwise, $F_{sd,w}^{ij} = 0$.

The objective (3-1) maximizes the total number of established lightpaths in the network. Equation (3-2) is the flow balance constraint at each node, which ensures that a lightpath entering a node on a particular wavelength leaves the node on the same wavelength. Equation (3-3) ensures that the number of established lightpaths is less than the number of requested connections. Constraint (3-4) dictates that each wavelength channel can only be assigned to one lightpath.

In a path-based formulation, for each lightpath request, a pre-calculated *candidate path set* (CPS) is provided. The route of a request can only be chosen from the corresponding CPS. When the pre-calculated CPS covers all the possible routes for a given request in a given network, the ILP solution is guaranteed to find an optimal solution for the SLE problem. An example of a path-based ILP formulation [28] for the MLE problem is given below.

Path-based ILP formulation: (PILP)

Objective:

$$\text{Maximize } \sum_{s,d,w,i} l_{sd,i}^w \quad (3-7)$$

Subject to the following constraints:

$$\sum_{r_{sd}^i \in C_{mn}} l_{sd,i}^w \leq 1, \forall (m,n) \in E, w \in W \quad (3-8)$$

$$\sum_{r_{sd}^i \in R_{sd}, w} l_{sd,i}^w \leq \lambda_{sd}, \forall (s,d) \in D \quad (3-9)$$

Here, R_{sd} donates the pre-computed candidate path set for node pair $(s,d) \in D$.

Also, we use r_{sd}^i to denote the i th route in R_{sd} . Let C_{ij} denote a set of candidate routes which use link $i \rightarrow j$. C_{ij} can be computed from R_{sd} for all the node pairs. Binary decision variable $l_{sd,i}^w$ is used to represent a lightpath: $l_{sd,i}^w = 1$ if there is a lightpath from s to d on wavelength w and that lightpath uses route r_{sd}^i ; otherwise, $l_{sd,i}^w = 0$. The objective (3-7) maximizes the total number of established lightpaths in the network. Equation (3-8) imposes the constraint that no two lightpaths can use the same wavelength on the same link. Equation (3-9) ensures that the number of established lightpaths is less than the number of requested connections.

The number of variables in the LILP grows in $O(|W| \times |E| \times |D|)$. The number of constraints in the LILP grow in $O(|W| \times |D| \times |N| + |W| \times |E|)$. The number of variables in the PILP grows in $O(\sum_{w,s,d} |R_{sd}|)$ and the number of constraints grows in $O(|W| * |E| + |D|)$. For a large network of which $|W|$, $|D|$, $|E|$ and $|R_{sd}|$ are large, both ILP formulations are virtually unsolvable. For this reason, *randomized rounding*

technique is used to find a near optimal solution when finding the exact solution is too computationally intensive [25][29]. In *randomized rounding*, the original ILP formulation is relaxed to a linear programming formulation by deleting those constraints that contain variables that can only assume integer values. After the linear relaxation is solved, each variable with a fractional value is rounded to an integer value in a random manner. Reference [29] shows that *randomized rounding* works quite well with the link-based ILP formulation for small networks for the MLE problem. Reference [25] shows that randomized rounding can be used for the MWU problem with good results, even for large networks. However, due to the inherent complexity of solving the linear relaxation for a large formulation, ILP approach is generally regarded as computational intensive.

3.1.2 Heuristic Solutions

The SLE problem can also be formulated as a *combinatorial optimization* problem. For this reason, heuristic algorithms, which are widely applied in solving various combinatorial optimization problems such as *genetic algorithms* (GA) and *tabu search* (TS), begun to surface as alternative solution techniques. Genetic Algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information. An implementation of a genetic algorithm begins with a population of typically random chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes that represent a better solution to the target problem are given more chances to reproduce than those

chromosomes that are poorer solutions. Generally, chromosomes reproduction is done by invoking a crossover function, where parts of two parent chromosomes are exchanged to form new next generation chromosomes. Recently, GA is used in some optical network related problems. In [30][31], different generic algorithms are proposed for the network design problem, of which the objective is the network cost. In [27], a genetic algorithm is proposed to solve the MWU problem; the simulation results show that it can work quite well for small networks. However, there is no easy way to extend the algorithm proposed in [27] to large networks without significantly increase the computational cost.

As described in [32], tabu search is “a meta-heuristic superimposed on another heuristic, the overall approach of tabu search is to avoid entrainment in cycles by forbidding or penalizing moves which will take the solution, in the next iteration, to points in the solution space which has been visited previously (hence "tabu")”. The tabu search begins with an initial “current” solution, and a neighborhood is constructed based on the current solution. The searching process goes from one neighborhood to another neighborhood to find a suitable next current solution. To avoid re-searching a neighborhood that has been searched recently, the method records recent moves in a data structure called tabu lists. This mechanism ensures that new regions of a problem solution space will be investigated with the goal of avoiding local minima and ultimately finding the desired solution. TS are widely used in such combinatorial problems as graph coloring [33], scheduling [34], etc. Recently TS has begun to be used in optical network related problems. In [35], a TS algorithm is used to find the optimal route for multicast sessions in an optical network. In [36], planning of optical ring networks is optimized by means of a TS algorithm. In [37], a TS

algorithm is designed to solve the RWA problem of scheduled lightpath requests, which is analogous to the MLE problem described here.

Though the SLE problem has been studied by a few researchers, we notice that, to tackle a SLE problem for a large network, most researchers adopt the two-step approach of [25]. In the two-step approach, the SLE problem is decoupled into two sub-problems, which are solved separately in sequence. First, the routes of the requests are found by means of an ILP based algorithm [25], evolutionary algorithm [27] or tabu search [37]. Secondly, once the route has been chosen for each request, wavelengths are assigned to each route in a manner that no two resulting lightpaths that need to share the same link use the same wavelength. In general, wavelength assignment is carried out by using existing graph-coloring algorithms. This two-step approach has the merit of significantly reducing the number of variables in each step, and thus can significantly improve the computation efficiency. However, since the route and wavelength assignment are chosen independently from each other, the optimality of the solution found is compromised. We notice that when TS is applied, the complexity of the algorithm is mainly governed by the neighborhood selection of the search process instead of the whole search space, which is proportional to the number of variables. In other words, if we consider routing and wavelength assignment jointly in TS, though the number of decision variables will be large, with carefully designed neighborhood construction, we can still achieve computational efficiency while maintaining the optimality. For this reason, in Chapter 4, we propose a TS algorithm based on this approach for solving the MLE and MWU problems that belong to the SLE problem category. Our simulation study for relatively large

networks shows that the results yielded by our proposed TS algorithm are almost as good as that yielded by the ILP solution and require much shorter running time.

3.2 Dynamic Lightpath Establishment (DLE) Problem

In DLE, lightpath requests are assumed to be initiated randomly by clients, and routing and wavelength assignment for the establishment of a lightpath in response to the request is based on the network state at the instant the request arrives. For a WRON network with E links and W wavelengths, the state S_i^t of link l_i ($0 < i < E$) at time t can be represented by a vector $(w_i^t(0), w_i^t(1), \dots, w_i^t(W-1))$. Here, $w_i^t(j) = 0$ if wavelength j of l_i is occupied by any existing lightpath at time t ; otherwise $w_i^t(j) = 1$. For a request that arrives at time t , routing and wavelength assignment involved finding a route $R = \{l_1, l_2, \dots, l_x\}$ from the source node, where the request is initiated, to the destination node, and some wavelength j such that $w_k^t(j) = 1$ for all $k = 1, 2, \dots, x$. If the RWA algorithm fails to find such a lightpath due to lack of resources at time t , the request is blocked. The commonly used objective of DLE is to minimize the request-blocking rate. Since the network state changes dynamically as and when lightpath requests are admitted and existing lightpath are torn down continually, the RWA algorithm needs to be executed in real-time.

It is shown in [38] that finding optimal results for the DLE problem is NP-complete; thus, heuristic algorithms are needed.

Routing and wavelength assignment in DLE can be solved in a single step. In [39], a *full adaptive algorithm* (FAA) is proposed to solve the routing problem and the

wavelength assignment problem jointly in a single step. In the FAA, firstly, a layered graph is formed from the given network topology (Figure 3.1). For a WRON network with W wavelengths, the layered graph consists of W identical sub-graphs (wavelength planes), which conforms to the physical topology. For the source node and destination node of the incoming request, auxiliary nodes are added to connect each wavelength plane. If j^{th} wavelength in a fiber link is occupied by any existing lightpath, the cost of the corresponding link in the j^{th} wavelength plane is set to infinite so that it will not be considered by the routing algorithm. After the above-described graph manipulation, the RWA problem can be solved by applying any shortest path algorithm to find a path from the auxiliary source node to the auxiliary destination node on the layered graph. As can be seen from the above description, the routing problem and wavelength assignment problem are transformed into a routing problem on the layered graph. The simulation results reported in [39] show that FAA outperforms the fix-alternate routing algorithm.

Figure 3.1 Graph conversion in FAA

Despite of the advantage described above, FAA has its own problems:

- 1) The FAA algorithm needs each OXC to maintain up-to-date global link state of the network so that accurate layered graphs can be constructed. For each lightpath accepted by the network, the states of the affected links should be updated to reflect the change in the number of wavelengths available. Such updating may significantly increase network control traffic due to frequent flooding of link state information by network nodes, which in turn imposes a heavy control overhead in the control plane.
- 2) The computation cost of FAA is high when the number of wavelengths, W , is large. This is easily seen because the physical topology has to be duplicated W times to generate the W -layer graph. The computation complexity will increase by at least W times [40].

Because of the complexity of the above combined approach in solving the DLE problem, the most widely used approach is to decouple the RWA problem into two separate sub-problems: the routing sub-problem and wavelength assignment sub-problem.

3.2.1 Routing

3.2.1.1 Fixed Routing

This is the most straightforward approach to solving the routing problem. In fixed routing, each source and destination node pair is assigned a predetermined route, which can either be calculated using a standard shortest path algorithm, say Dijkstra's algorithm, or selected in a manner that balances the load evenly across all the links in the network. When a lightpath request reaches the network, the ingress node will attempt to find a common wavelength along the fixed route. If it fails to do so, the request is blocked. Fixed routing does not require the maintenance of global link states, thus it has the least control overhead. However, ignoring the network state and failing to consider the availability of wavelengths in different links usually lead to a high blocking rate compared to other schemes.

3.2.1.2 Fixed-Alternate Routing

This is an improvement of the fixed routing approach. In fixed-alternate routing, each node maintains a routing table, which contains a set of predetermined fixed routes to each destination node in the network [41][42][43]. In [42], a set of link-disjoint routes is calculated offline and ordered according to the route's hop count. On receiving a lightpath request, the network will first attempt to set up a lightpath on the

first route. If a common wavelength along the route can be found, the corresponding request is accepted; otherwise the second route is examined. If a lightpath cannot be found after all the routes have been tried, the request is blocked. With a larger freedom in route selection, the fixed-alternate scheme can significantly outperform the fixed routing scheme in term of blocking. Similar to fixed routing, the fixed-alternate routing scheme proposed in [42] does not require the maintenance of global link states and is easy to implement.

In [43], the selection of a route not only depends on the order in the predefined route set but also the network state. Two routing schemes are proposed in [43]. The first scheme is *fixed-path least-congestion* (FPLC) routing. In FPLC routing, a set of link-disjoint paths are pre-calculated and ordered according to its hop count for each node pair in the network. When a lightpath request arrives, the source node will send out probing messages separately along each path to the destination node. The probing messages will collect the wavelength availability information of the path it travels. When the destination node receives all the probing messages, it knows how many wavelengths are still free for each path. Then, it chooses the path with the largest number of idle wavelengths to set up the lightpath. Tie is broken by selecting the path with the lowest index number. When applying FPLC to a large network, it may experience problems of large control overhead caused by probing messages and long setup delay because of the message transmission and processing delay. For this reason, another scheme based on FPLC, called FPLC-N(k), is proposed. In this scheme, the wavelength availability information for the first k links along the candidate paths are gathered and routing decision is made. The simulation result reported in [43] shows that FPLC can significantly outperform other fixed-alternate algorithms.

3.2.1.3 Adaptive Routing

In adaptive routing, the route is selected dynamically based on the current network state. Due to its consideration of current network resource information, adaptive routing algorithms normally outperform the fixed and fixed-alternate routing algorithms at the expense of increased computation complexity and control overhead in the network.

In [38], global link state information, such as wavelength availability information on each link, is assumed to be available to each node in the network. Different RWA algorithms are proposed in [38] as follows:

- 1) PACK: Wavelengths are sequenced in descending order according to their utilization in the network. For example, if wavelength λ_1 is used on five links by the existing lightpaths in the network, and wavelength λ_2 is used on three links by the existing lightpaths, we consider wavelength λ_1 to be more utilized compared to wavelength λ_2 . In this set of two wavelengths, the order will be λ_1 followed by λ_2 . The algorithm will attempt to route the request on the first wavelength in the ordered set of wavelengths. If successful, the lightpath is set up; otherwise, it continues with the next wavelength in the ordered set.
- 2) SPREAD. In this algorithm, the wavelengths are sequenced in ascending order according to their utilization in the network. This algorithm will achieve a near-uniform distribution of the lightpaths

over all wavelengths. The algorithm will attempt to route the request on the wavelength in the arranged sequence.

- 3) RANDOM. The wavelength set is sequenced randomly. The algorithm will attempt to route the request on the wavelength in sequence.
- 4) EXHAUSTIVE: The algorithm searches all the wavelengths to find the shortest path. This algorithm is similar to the FAA algorithm of [39] but does not carry out graph conversion.
- 5) FIXED: The wavelengths are sequenced in a fixed manner and does not change when network state changes. The algorithm will attempt to route the request on the wavelength in sequence.

Simulations in [38] show that the EXHAUSTIVE and PACK algorithms can achieve almost the same blocking rate, and are better than all the other algorithms considered. PACK algorithm is preferred over EXHAUSTIVE algorithm since the former is more computationally efficient. When a suitable path is found, PACK algorithm may stop before all the wavelengths are searched while EXHAUSTIVE algorithm always needs to search all the wavelengths. FIXED algorithm can achieve a performance nearly as good as PACK algorithm since FIXED tends to assign most requests to the first available wavelength; thereby approximating PACK. Compared with PACK, FIXED is much simpler in implementation since it does not need to re-sort the wavelength set when network state changes.

Since the algorithms proposed in [38] are based on the assumption that global link states need to be accurately maintained, it suffers the same problem faced by FAA.

Hence, these algorithms are more suitable for networks in which lightpaths are fairly static and do not need to be set up and torn down frequently.

In [44], adaptive *alternate-link routing* algorithms based on local state update are proposed. A notable advantage of the alternate-link routing algorithm is that it does not require the maintenance of global link state information in the network, and thus avoids the high control overload caused by link state flooding.

In *alternate-link routing*, each node maintains a routing table. The routing table has an entry for each destination, which stores one or more alternate outgoing links that can be used to reach the destination. Each node maintains the wavelength availability information of each outgoing link. When a connection request reaches a node, the node chooses an outgoing link based on certain policy. Two policies are proposed and studied in [44]. The first one is shortest-path-first. In this policy, the outgoing link that leads to the shortest path to the destination will be checked for wavelength availability. When there is no ‘available’ wavelength in that link, the outgoing link that leads to the next shortest path to the destination will be checked. Here, the term “available” wavelengths refer to a set of wavelengths that are available on all of the links that constitute the path from source to destination. The second policy is known as least-congested. In this policy, the outgoing link that has the maximum number of ‘available’ wavelengths is chosen.

Upon choosing an outgoing link, the node forwards the connection request to the next node. The connection request is routed in a hop-by-hop manner until it reaches the destination or is blocked if at certain intermediate node all the outgoing links do not have any available wavelength. When the connection request reaches the

destination node, a wavelength is selected from the set of available wavelengths and the lightpath is set up. To prevent routing loops, revisiting a node that has already been visited by the connection request is not allowed.

Simulation results reported in [44] show that the proposed algorithms outperform the fix-alternate routing algorithm at light load but lose out at heavy load.

3.2.2 Wavelength Assignment

Wavelength assignment algorithms deal with assigning a wavelength to a connection request. It can either be performed after a route has been selected or carried out simultaneously when a route is being searched.

The following algorithms have been proposed for single-fiber WRONs in the past:

- 1) Random Wavelength Assignment (RANDOM): A set of available wavelengths are first determined by searching each link of the given route; then a wavelength is randomly selected from the set of available wavelengths [41].
- 2) First-Fit: All wavelengths are indexed. In the process of searching for an available wavelength, the wavelength with the smallest index is considered before a higher indexed wavelength. The search process stops when the first available wavelength is found [28]. The idea behind this scheme is to route as many lightpaths to lower indexed wavelengths as possible so that higher indexed wavelengths are less utilized and are more likely to accommodate requests with long paths.

- 3) Most-Used: The algorithm tries to select the most used wavelength to set up a lightpath [38]. Compared with RANDOM and First-fit, the Most-used algorithm requires the network to maintain the wavelength utilization information, and this incurs extra control overhead.
- 4) Wavelength Reservation (WR): In WR, the network will reserve a given wavelength on a specific link for certain connection requests, usually a multi-hop connection [45]. This scheme reduces blocking for long connections at the expense of single-hop connections.
- 5) Protection Threshold (PT): A threshold in term of the number of idle wavelengths on a link is set. When the number of available wavelengths on a link is smaller than the threshold, no single-hop connection is allowed on that link [45].

The results reported in [5] show that the Most-used scheme has the best performance, followed by First-fit and then Random.

3.3 Virtual Topology Design (VTD) Problem

In *virtual topology design* (VTD), the physical topology of WRON network and the traffic matrix, which specifies the average IP traffic between every node pair in the physical topology, are given. A lightpath created in the optical layer defines a logical link in the IP layer. VTD thus involves allocating logical links (lightpaths) among network nodes in a global fashion while optimizing certain performance

objective. The most studied objectives are related to network congestion and average packet delay.

3.3.1 ILP Formulations

The optimal solution of VTD is usually found by means of *mixed integer linear program* (MILP) formulation. The following formulation was originally proposed in [9] and rewritten by us to make it consistent with other formulations presented in this dissertation. Similar formulations for different objectives are described in [8][46].

Notation:

Given:

λ^{sd} : The arrival rate of packets at source node s that are destined for node d .

F_{mn} : $F_{mn}=1$ if a fiber link exists in WRON between nodes m and n ; otherwise, 0

W : Number of wavelengths per fiber link.

T : Number of transmitters and receivers of a node. Note that all nodes in WRON are assumed to have the same number of transmitters and receivers.

Variables:

Logical link variables:

b_{ij} : Number of logical links between nodes i and j .

Wavelength assignment variables:

L_{ij}^w : $L_{ij}^w=1$ if a logical link between nodes i and j uses wavelength w ; otherwise

$$L_{ij}^w=0;$$

$L_{ij}^w(mn)$: $L_{ij}^w(mn)=1$ if a logical link between node i and j uses wavelength w and is routed through fiber link (m,n) ; otherwise, $L_{ij}^w(mn)=0$.

Traffic intensity variables:

λ_{ij}^{sd} represents the traffic intensity of logical link (i,j) for the traffic between node pair $s-d$.

Objective :

$$\text{Minimize } \lambda_{\max} \quad (3-10)$$

Subject to the following constraints:

$$\sum_j b_{ij} \leq T, \forall i \quad (3-11)$$

$$\sum_j b_{ji} \leq T, \forall i \quad (3-12)$$

$$\sum_w L_{ij}^w = b_{ij}, \forall (i, j) \quad (3-13)$$

$$L_{ij}^w(mn) \leq L_{ij}^w, \forall (i, j), (m, n), w \quad (3-14)$$

$$\sum_{ij} L_{ij}^w(mn) \leq 1, \forall (m, n), w \quad (3-15)$$

$$\sum_w \sum_n L_{ij}^w(mn) \times F_{mn} - \sum_w \sum_n L_{ij}^w(nm) \times F_{nm} = \begin{cases} b_{ij}, n = i \\ -b_{ij}, n = j \\ 0, n \neq i, n \neq j \end{cases} \quad \forall (i, j), m \quad (3-16)$$

$$\sum_{sd} \lambda_{ij}^{sd} \leq \lambda_{\max}, \forall(i, j) \quad (3-17)$$

$$\lambda_{ij}^{sd} \leq b_{ij} \times \lambda^{sd}, \forall(i, j), (s, d) \quad (3-18)$$

$$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd}, s = i \\ -\lambda^{sd}, d = i \quad \forall(s, d), i \\ 0, s \neq i, d \neq i \end{cases} \quad (3-19)$$

Equations (3-11) and (3-12) constrain the logical topology to the given logical degree. Equations (3-13) to (3-16) will calculate the route and wavelength in the optical layer for logic link (i, j) . Equation (3-17) defines network congestion as the maximum total traffic for all logical links. Equation (3-18) ensures that the traffic on logical link (i, j) between source and destination node pair $s-d$ is upper-bounded by the total flow of traffic λ^{sd} . Equation (3-19) describes the flow conservation constraint at each node for traffic between s and d .

3.3.2 Heuristic Algorithms

The VTD problem is NP-hard [9]; thus a heuristic algorithm is needed to find a near-optimal solution.

In [9], a heuristic algorithm based on linear relaxation of the MILP formulation is developed. LP-relaxation formulation of the MILP is solved first. Then, the relaxed integer variable with fractional value is rounded to an integer value according to certain policy. After the rounding process has ended, the logical topology and the route of lightpaths are decided. Now, graph-coloring algorithm is used to decide the number of wavelengths used.

Since the MILP is very complex when the network is large, even LP-relaxation may become insolvable. For this reason, more researchers propose much simpler heuristics for large networks [8][46][47]. Though their algorithms may differ from each other in details, they capture essentially the same features. In those algorithms, the traffic demands are first sorted according to certain policies. Depending on the sorted order, a lightpath is assigned to each demand using a simple wavelength assignment algorithm, such as First-fit. Virtual topology is built up gradually with each demand successfully routed. When there is no more resource in the optical layer to set up more lightpaths, the remaining traffic demand is routed over the existing virtual topology.

3.4 Dynamic LSP Provisioning (DLP) Problem

In IP-over-optical networks, the IP layer is built directly on top of the optical layer, and the lightpath connections in the optical layer defines the logical links (virtual links) between the IP network nodes. With GMPLS, an IP traffic demand, which needs to be forwarded over the same path and treated in the same manner by the network, is regarded as an LSP request with certain bandwidth requirement. The DLP problem deals with routing an LSP request dynamically in an IP-over-optical network. Here, the term “dynamically” means the LSP requests will reach the network at random timing epochs, and for each new LSP request that arrives, the ingress LSR needs to make a routing decision in real-time based on network state information it maintains. A widely studied objective of the DLP problem is to minimize the blocking rate of LSP requests.

The LSP request can be routed solely in the IP layer, however, from resource utilization point of view, the flexibility of routing the LSP request in either the IP layer, optical layer or both taking into consideration the available resources in both layers can achieve much better network performance. For this reason, different routing algorithms that adopt the two-layer routing approach are proposed [48][49][50][51],

In [48] and [49], similar *Integrated Routing Algorithms* (IRAs) are proposed. An IRA is applicable under the assumption of the peer model [4] of IP-over-optical networks. For the peer model, each LSR in the network is required to maintain up-to-date global link states of both the IP and optical layers. Upon the arrival of an LSP request, the ingress LSR will form a routing topology (Figure 3.2) based on the global link state information it maintains. The optical layer is converted into a layered graph similar to what is described in [48] so that routing and wavelength assignment in the optical layer is reduced to a routing problem. The IP layer will only consist of those existing virtual links that have larger residue bandwidth than the required bandwidth of LSP requests. After the graph manipulation, any standard shortest path algorithm can be used to find an LSP route.

Figure 3.2 Routing topology in IRA

The IRA has the merit of optimally utilizing the spare resources of both layers; however, it may lead to high control overhead since each LSR is required to maintain the state of the optical links. When a lightpath is set up or torn down in the optical layer, optical link state update messages will flood both the IP and optical layers.

IP-layer-first and optical-layer-first routing algorithms are proposed in [50][51]. Both algorithms are applicable under the assumption of the overlay model [4] of IP-over-optical networks. For the overlay model, the LSR only needs to maintain IP layer link states, and can issue lightpath requests to the optical layer so that new virtual links can be created. In the IP-layer-first algorithm, the LSP request is first routed in the IP layer; only if the IP layer does not have enough capacity to accommodate it, will the request be directed to the optical layer. The optical layer will try to establish a lightpath between the corresponding OXC node pair. If the attempt fails, the LSP request will be blocked. In contrast, for the optical-layer-first strategy,

the LSP request will always result in the creation of a new lightpath in the optical layer first. If the lightpath can be established successfully, it is used to serve the LSP; otherwise, the request is directed to the IP layer for handling. If both the layers are unable to fulfill the request, the request is blocked.

Clearly, integrated routing will outperform the IP-layer-first or optical-layer-first algorithms [50], since integrated routing will utilize the resources of both layers while the other two can only use the resources of either the IP layer or the optical layer. However, since the overlay model is more practical in near term in term of implementation feasibility compared to the peer model [4][52], it is attractive to find new algorithms for the overlay model which can perform almost as good as integrated routing for the peer model. For this reason, in Chapter 6, we propose an enquiry mechanism which works for the overlay model and allow the IP layer to obtain spare resource information of the optical layer. Also, in that chapter, the effect of using different dynamic RWA algorithms in the optical layer on the complexity of the enquiry process is discussed. Based on the proposed enquiry process, two routing algorithms suitable for the overlay model are described in Chapter 7. Simulation results show that their performances are almost as good as that of integrated routing even though they are much simpler.

3.5 Other Relevant Research Work

Apart from those solution techniques for RWA problems which are reviewed in Sections 3.1 to 3.4, we observe that some important research works that involve new methodologies emerged in around 2004 towards the end of the study described in this dissertation. In order for the readers to appreciate the contributions of this dissertation, these works are reviewed here.

In [59], the authors study static RWA for multiple fiber networks. The optical network is assumed to be connected by links which consist of multiple fibers. The objective of the study is to maximize the carried traffic of connections. A Path based ILP formulation, which has the same number of variables and constraints as the PILP formulation discussed in Section 3.1, is proposed in the paper. As discussed in Section 3.1, for a large network, even the linear programming relaxation (LP-relaxation) of such an ILP will not be solvable since the number of variables and constraints are simply too large. For this reason, the authors propose another LP-relaxation scheme in which the number of wavelengths per link is assumed to be 1 so that the number of constraints and variables can be significantly reduced. The authors prove that the upper bound of the reduced LP-relaxation scheme can achieve the same upper bound as the LP-relaxation of the original ILP formulation. In addition, the authors propose a greedy algorithm where in each iteration, only the reduced LP-relaxation problem needs to be solved. The proposed greedy algorithm is proved to be able to find a solution within a factor of $(1 - 1/e)$ of the original ILP formulation's optimal value. The authors show that the proposed greedy algorithm works well for small networks of 20 nodes. However, we expect the proposed algorithm to be computationally intensive for large networks.

In [60][61], Lagrangean Relaxation based solutions are provided for SLE problems. In [60], multi-granularity optical WDM networks are studied. In multi-granularity optical WDM networks, an OXC node is assumed to be capable of switching wavelengths, fibers or both fibers and wavelengths. Comparing to a network where only wavelengths are switched, this additional freedom compounded the complexity of the SLE problem. The objective of the study is to minimize the

most congested link utilization. In [61], the optical networks studied are assumed to have limited capability of wavelength conversion and the objective is to minimize the penalty cost for rejected lightpath requests. Different ILP formulations are proposed in [60] and [61] to form the bases of the proposed heuristic algorithms. Then Lagrangean Relaxation is adopted for finding the upper and lower bounds of the original ILP formulations. In Lagrangean Relaxation, complex ILP constraints are relaxed such that the relaxed problem can be decomposed into independent manageable sub-problems. Unlike the traditional Linear Programming Relaxation (LP-Relaxation) where integer variables of ILP are relaxed to non-integer values, in Lagrangean Relaxation, selected constraints of ILP are multiplied by Lagrangean multipliers and then summed with the original objective function. The ILP problem is then transformed into a simplified dual problem which can find a lower or upper bound of the original problem. In [62], comparing with LP-relaxation, Lagrangean Relaxation is shown to be able to provide tighter bounds and shorter computation time for the optimal values of objective functions in many instances. In both [60][61], heuristic algorithms based on Lagrangean Relaxation are proposed and results are compared with the LP-relaxation approach. Both papers show that their Lagrangean Relaxation based algorithms can outperform the LP-relaxation based algorithm in terms of performance and computation time for small and large networks. However, we observe that even the Lagrangean Relaxation approach is more computational efficient than LP-relaxation, it is still computational intensive for large networks. In [60], the authors show that for a network of 61 nodes and 148 bi-directional links, the Lagrangean Relaxation based algorithm take hours in finding a near optimal result.

Chapter 4

Tabu Search Solutions for the SLE Problems

In this chapter, we describe a *tabu search* (TS) algorithm for solving the SLE problem. First, we give a short general introduction to TS algorithms. Then, we describe a tabu search algorithm for a well-known SLE problem; i.e., MLE problem. Finally, we describe how to make use of the proposed MLE TS algorithm to solve another important SLE problem; i.e., MWU problem.

4.1 Generic Tabu Search Algorithm

Tabu search (TS) [53] is as an iterative heuristic algorithm suitable for solving *combinatorial optimization problems*, in which the set of feasible solutions are discrete and the goal is to find the best possible solution.

Figure 4.1 Solution space in local search

TS can be regarded as a generalization of the *local search* heuristic. Local search is a simple iterative algorithm that has been applied with much success for a variety of hard combinatorial optimization problems. We use a directed graph $G = (\Omega, E)$ (Figure 4.1), where E is the set of directed links and Ω is the set of nodes in the graph, to illustrate the solution space being searched in a local search. Each node $\Gamma \in \Omega$ represents a feasible solution and Ω represents the set of all feasible solutions. A directed link $\Gamma \rightarrow \Gamma'$ indicates that solution Γ' is a neighbor of Γ . Each solution Γ has an associated neighborhood $\aleph(\Gamma) = \{\Gamma' \mid (\Gamma \rightarrow \Gamma') \in E, \Gamma' \in \Omega\}$. The neighborhood relation is assumed symmetric, in other words, if there exist link $\Gamma \rightarrow \Gamma'$ in the graph G , then there exists link $\Gamma' \rightarrow \Gamma$. The operation by which a solution $\Gamma' \in \aleph(\Gamma)$ is reached from Γ is called a *move*. A cost function $C(\Gamma)$ is formulated to evaluate the fitness of the solution in relation to the objective of the optimization problem. The heuristic starts at some initial feasible solution $\Gamma \in \Omega$ and a search is conducted in $\aleph(\Gamma)$ to find a solution Γ' with $C(\Gamma') < C(\Gamma)$ in the case of a minimization problem. If Γ' is found, it is dubbed as current solution and the algorithm continues to search the neighborhood $\aleph(\Gamma')$ for the next current solution. The local search heuristic stops when the search process cannot find a better solution in the neighborhood $\aleph(\Gamma)$. The final Γ is known as *local optimum*, since it is at least as good as or better than any solutions in its neighborhood. The disadvantage of the local search is that, in most cases, such a local optimum will not be a global optimum; in other words, it usually cannot find the solution $\Gamma \in \Omega$, which is the best $C(\Gamma)$ over all other possible solutions.

Tabu search applies a guidance mechanism, which is designed to improve the effectiveness of choosing good solutions from the current neighborhood, to go beyond the locally optimal stopping point of local search. In an iteration of the TS algorithm, multiple trial solutions are examined in the neighborhood, and the solution with the best $C(\Gamma)$ among all other trial solutions are selected as the next current solution. The reason for selecting the best solution among the trial solutions is based on the assumption that a good solution is more likely to be optimal or near-optimal. Note that in local search, the selection of the next current solution is governed by the requirement that $C(\Gamma') < C(\Gamma)$. If such a solution cannot be found, the local optimal is reached and the algorithm stops. In TS, the selection of the next current solution is based on the comparison among the trial solutions in the neighborhood, thus a solution that is worse than the current solution may be chosen. This mechanism ensures that the search process continues after the local optimum is hit, in a “hill-climbing” way. However, since the escape from the local optimum may involve an ascend progress where Γ' with $C(\Gamma') > C(\Gamma)$ is chosen as next current solution, it is quite possible that the search process may return to the same local optimum in later iterations. To prevent cycles, TS makes use of a memory-based cycle prevention mechanism.

A short-term memory component is used in TS to maintain selected history of states encountered during the search process. The most commonly used short-term memory is called *recency-based* memory, in which, solutions that were encountered in recent searches are maintained. Selected attributes that occur in recently visited solutions are labeled as *tabu-active*, and solutions that contain the tabu-active attributes, or particular combinations of those attributes, are classified as *tabu*. With

tabu defined, tabu solutions are excluded from the original neighborhood $\aleph(\Gamma)$ of the current solution Γ , and a new neighborhood $\aleph^*(\Gamma)$, which is a subset of $\aleph(\Gamma)$, is formed. The purpose of excluding recently visited solutions from the neighborhood is based on the observation that revisiting a recently encountered solution will trap the search process into a cycle. For example, in the current iteration, a move is conducted and the search process is directed from Γ_1 to $\Gamma_2 \in \aleph(\Gamma_1)$. Clearly, in the next iteration, the solution Γ_1 should be excluded from $\aleph(\Gamma_2)$ to avoid a return to a search state already encountered.

Storing complete tabu solutions need an enormous amount of space and processing time; hence, selected attributes of moves are used to replace complete solutions as tabu. A list-like data structure known as *tabu list* is used to store these attributes that represent tabu solutions. In the tabu list, the attributes of κ most recent moves are stored. The size κ of the tabu list reflects how long a period of recent search history is maintained, and is generally known as *tabu tenure*. The tabu list is maintained throughout the whole search process. When a new solution is accepted in the search process, selected attributes of moves leading to that solution is added as a tabu at the end of the list. When the size of the list grows larger than the tabu tenure after a new tabu is added, the oldest (the *head*) entry in the tabu list is deleted.

Tabu list is used to prevent the search process from cycling back to a solution recently visited. However, since only the move attributes but not the exact solution are maintained, some solutions that have not been visited before in the search process but have the same move attributes may also be excluded by the tabu list. For example, in a TS application in which a move is defined as adding a link or deleting a link in

the current solution, in iteration 1, link l_1 is added to solution Γ_1 and a new current solution Γ_2 is formed. Since the algorithm does not maintain the exact solution of Γ_1 in order to reduce the complexity, we have to define the move, which deletes link l_1 , as a tabu because deleting l_1 is likely to cause a return to solution Γ_1 . Now, before the tabu of deleting l_1 expires, a solution that is different from Γ_1 but needs to delete l_1 from the current solution is forbidden. To overcome this problem, Aspiration criteria are used to override Tabu list if a tabu move has potential to reach better solution. Several aspiration criteria are introduced in [54], which include *regional aspiration by objective*, and *aspiration by search direction*. For *regional aspiration by objective*, the solution space Ω is divided into several regions. If a move can find a solution that is better than the best solution in the current solution region, this move is deemed acceptable and can override the Tabu list. *Aspiration by search direction* favors moves that have previously led to better improvement in costs. If a move can lead to a better solution and its reversal move also leads to a better solution, the tabu rule is overridden. For example, in a TS application, a move is defined as adding a link or deleting a link in the current solution. In iteration 1, link l_1 is added to solution Γ_1 and a better solution Γ_2 is formed. Assume a tabu tenure of 10; any move within the next 10 iterations that deletes l_1 will be regarded as a tabu. Further assume that in iteration 7, a better solution can be reached by deleting l_1 . With *aspiration by search direction*, the move of deleting l_1 is no longer regarded as tabu since both adding l_1 and deleting l_1 result in better solutions. The simplest and most widely used criterion is the *global aspiration by objective*: When a move can yield a solution that is better than the best solution obtained so far, it will no longer be regarded as a tabu. This is also known as *best solution aspiration criterion*.

A set of pseudo codes for a generic tabu search algorithm is given below [54]:

Begin of main //TS algorithm

Construct an initial feasible solution $\Gamma_{ini} \in \Omega$;

$\Gamma \leftarrow \Gamma_{ini}$;

Initialize *tabu list* and *aspiration level*;

While (Iteration number < given fixed number) **do**

Begin

Generate neighborhood solutions $V^* \in \aleph(\Gamma)$;

Find best $\Gamma' \in V^*$;

If (the move Γ to Γ' is not in the *tabu list*) **then**

Accept *move* and update best solution;

Update *tabu list* and *aspiration level*;

$\Gamma \leftarrow \Gamma'$;

Increment iteration number;

Else

If ($C(\Gamma') < \text{aspiration level}$) **then**

Accept *move* and update best solution;

Update *tabu list* and *aspiration level*;

$\Gamma \leftarrow \Gamma'$;

Increment iteration number;

Endif

Endif

End of While

End of Main

The procedure starts from an initial feasible solution $\Gamma_{ini} \in \Omega$. In a real-world application, the neighborhood associated with a solution is generally huge, and thus calculating the whole neighborhood is impractical (computationally expensive). A small collection of trial solutions of neighborhood $V^* \in \aleph(\Gamma)$ is sampled with the

requirement $|V^*| < |\mathcal{N}(\Gamma)|$, and from V^* , the best solution $\Gamma' \in V^*$ is chosen as the next solution, even if it is worse than the current solution, say $C(\Gamma') > C(\Gamma)$ in case of a minimization problem. The move from Γ to Γ' is accepted only if it is not in the tabu list; or if it is in the list but $C(\Gamma')$ is less than the *aspiration level* (in case of a minimization problem). When the best solution aspiration criterion is adopted, the aspiration level is equal to the best solution found so far.

More detailed description of generic tabu search algorithms and more sophisticated techniques used in TS algorithms, such as long-term memory utilization, path re-linking, strategic oscillation etc. can be found in [53][54]. Though using more sophisticated techniques may result in better performance, it generally complicates the algorithm. In what follows, we propose a simple TS algorithm, which can be used to solve both the MLE and MUS problems. Our proposed TS algorithm will only utilize the basic TS idea described above. Our simulation results show that the proposed algorithm works very well in finding an optimal or near optimal solution while keeping the computational cost low.

4.2 MLE Problem with TS Solution

The MLE problem can be formally defined as follows:

Given a graph $G(E, N)$ and $|W|$ number of wavelengths per fiber link, where E is the set of links and N is the set of node in the graph, W is the set of wavelengths that a link supports, and $|W|$ denotes the number of wavelengths in W . Also given is a set of

lightpath requests $\Delta = \{R_1, R_2, \dots, R_{|\Delta|}\}$. Find a subset $\Lambda \subset \Delta$, which fulfills the following constraints:

- (MLE_C1): For each $R_i \in \Lambda$, a lightpath should be set up in the given network.
- (MLE_C2): Established lightpaths will strictly follow both the *wavelength continuity constraint* (WCC) and the *lightpath conflict constraint* (LCC). In other words, a lightpath should use the same wavelength along its route (WCC) and for any two lightpaths established, they cannot use the same wavelength if they share any links (LCC).

The objective is then to maximize the cardinality $|\Lambda|$.

The proposed TS algorithm will search for an optimal solution that consists of the routes and the wavelengths to be assigned. When routing and wavelength assignment are considered jointly, the solution space is much larger. However, with a carefully constructed neighborhood, we can still achieve computational efficiency without compromising optimality.

To apply the TS approach in solving the MLE problem, we use variable L_i to represent a lightpath assigned to a request $R_i \in \Delta$. In our proposal, a candidate path set P_i is pre-calculated for each request R_i using the k -shortest paths algorithm in [55] and the set of lightpaths are ordered in accordance with their hop counts. L_i takes the form of (p_i^x, w) where p_i^x is the x^{th} path in P_i and w is the w^{th} wavelength in W .

Two sets of decision variables are maintained by the TS algorithm, namely, *assigned variable set* (AVS) and *waiting variable set* (WVS). When a variable is assigned a value, it is added to AVS, and all the other variables which are waiting for a value to be assigned are stored in WVS.

A *move* is defined in the proposed TS algorithm as follow:

Assign a value (lightpath) to a variable $L_x \in WVS$ and move it from WVS to AVS. When $AVS \neq Null$, the newly added L_x may be in conflict with a set of existing variables in AVS, say $\{L_y, \dots, L_z\}$. In this case, we clear the values of $\{L_y, \dots, L_z\}$ and move $\{L_y, \dots, L_z\}$ from AVS to WVS.

In each iteration, a move operation is carried out. The AVS always maintains a set of variables. Each variable is assigned a specific lightpath in such a way that all the lightpaths assigned are not in conflict with each other. Physically, AVS represents a set of requests that can be simultaneously fulfilled by the network. Since the goal of MLE is to find the maximum number of requests that can be fulfilled, it is equivalent to finding the AVS with the largest number of variables in our TS algorithm. We use $CAVS$ to denote the number of variables in AVS.

Clearly, in the next current solution selection process, a *move* with the least number of variables deleted from AVS is preferred among all the possible moves since it results in an AVS with a larger $CAVS$. To evaluate the fitness of a move, a cost function $CMOVE$ is defined as the number of variables added to AVS minus the number of variables deleted from AVS. For example, if five variables are deleted

from the AVS as a result of a move, the cost of that move is $1 - 5 = -4$, since a move will always add a variable to the AVS.

The proposed TS algorithm begins with an initial solution of $WVS = \Delta$. In each iteration, multiple trial moves, which are based on WVS, are examined. If a trial move is not prohibited by the tabu list or that the tabu list is not overruled by the aspiration criteria, the move is regarded as a *legitimate move*. Among all the legitimate moves, the one with the maximum $CMOVE$ is accepted in that iteration and the corresponding updates of WVS, tabu list and $CAVS$ etc. are carried out. If there are ties in $CMOVE$, the move associated with the assignment of the lowest indexed wavelength to a variable is chosen. The idea behind this arrangement is to pack as many lightpaths as possible in wavelengths with low indices so that high index wavelengths can be reserved for longer lightpaths. A move of negative cost is acceptable in an iteration to provide “hill-climbing” capability so that local optimal can be escaped. The algorithm will stop when certain stop condition is met.

4.2.1 Trial Move Selection

A move is decided by two factors: (1) The variable selected from WVS, which defines which variable will be added to AVS; (2) the value assigned to the selected variable, which decides which variables in AVS will be deleted. The cardinality of all the possible moves for the current solution is then equated to $\sum_{L_x \in WVS} |P_x| \times |W|$, where $|P_x| \times |W|$ is the cardinality of the value domain of variable L_x . Clearly, examining all the possible moves when $|WVS|$ is large will be time-consuming. For this reason, the proposed TS search algorithm is carried out in two phases.

In Phase 1, only a small part of all the possible moves is examined. A variable is selected from WVS, and all the possible values for that variable are searched. In other words, there are only $|P_x| \times |W|$ trial moves examined in each iteration. The purpose of Phase 1 is to quickly reduce the size of WVS and lead the search process to a searching area, which may contain optimal or near-optimal results.

In Phase 2, after the size of WVS is reduced, all the possible moves are examined. Unlike Phase 1, this phase will search thoroughly all the neighbors of the current solution to find an optimal or near optimal solution.

4.2.2 Tabu List

Tabu list is used in the TS algorithm to prevent cycling in the search process. Since storing an exact solution requires ample memory space and is complicated, normally attributes of moves instead of exact solutions are maintain. In our proposed algorithm, the attributes associated with a move are classified into two categories: (1) the added variable L_x in AVS; (2) the deleted variables $L_y, \dots, L_z$ from AVS. There are three possible ways to construct a tabu list: (1) Define a newly added variable as a tabu and store it in the tabu list (adding tabu list). Then, in subsequent iterations, a move, which may delete a variable stored in the tabu list, is prohibited. (2) Define a newly deleted variable as a tabu and store it in the tabu list (deleting tabu list); then, in subsequent iterations, a move, which may add a variable stored in the tabu list, is prohibited. (3) Maintain both the adding tabu list and deleting tabu list; then, in subsequent iterations, a move that will add a variable stored in the deleting tabu list or delete a variable stored in the deleting tabu list is prohibited. We have tried three kinds of tabu list in our simulation, and the results show that all of them have almost

the same performance in terms of the optimality of the final solution found and convergence speed. We use the adding tabu list in our final algorithm since it is the easiest to implement.

4.2.3 Aspiration Criteria

Since one of our goals is to make the TS algorithm as simple as possible, we choose the simplest *best solution aspiration criteria*. We use *BCAVS* to maintain the best *CAVS* achieved so far. If a move, which is prohibited by the tabu list maintained, can result in an AVS with $CAVS > BCAVS$, the move is no longer regarded as tabu.

4.2.4 Stopping Condition

The proposed TS algorithm stops under two conditions: First, when $WVS = Null$; in this case, all the requests are fulfilled. Second, if *BCAVS* does not improve for *CYC* number of iterations, where *CYC* is a predetermined value; in this case, an optimal or near optimal result is found. The choice of *CYC* value is based on the following considerations: The proposed algorithm may be trapped in a local optimal for certain iterations before it manages to escape from the local optimal. If *CYC* is too small, the algorithm may stop at a local optimal. Since the algorithm will always continue for *CYC* number of iterations even after a global optimal solution has been found, a large *CYC* value will lead to unnecessary long computation time. The *CYC* value used in our proposed algorithm is decided by some experimental simulations of both small and large networks. We have used $|\Delta|/2, |\Delta|, 2 \times |\Delta|$ and $3 \times |\Delta|$ as the value for *CYC* and found that the value of $|\Delta|$ work quite well in term of the optimality of the final solution found and computation time.

4.2.5 Pseudo Codes for the Proposed TS Algorithm

We use a list to maintain the WVS and each of its element stores an unassigned variable. We do not maintain the AVS explicitly in the implementation since the AVS can be deduced from the operation $\Delta - WVS$. In an iteration, if variable L_x is moved from WVS to AVS and $L_y, \dots, L_z$ are moved from AVS to WVS, the WVS list is updated by deleting the corresponding element of L_x and inserting $L_y, \dots, L_z$ at the end of list.

We use a single-dimension array $A[]$ to maintain the current solution with element $A[i]$ storing the current value of L_i . Another single-dimension array $BA[]$ is used to store the best solution found so far. At the end of the algorithm, $BA[]$ stores the final solution of lightpath assignments for accepted requests. $BA[]$ is also used to initialize the WVS list in the Phase 2 subroutine. If $BA[x] = Null$, L_x is deemed to be an unassigned variable and should be stored in the WVS list.

$CAVS$ is initialized to zero, and built up in an incremental manner. After a move is selected, $CAVS$ is updated as $CAVS \leftarrow CAVS + CMOVE$. Here, the notation " $\leftarrow$ " denotes the operation of storing the right side value into the left side variable. If it results in $CAVS > BCAVS$, $BCAVS$ is updated as $BCAVS \leftarrow CAVS$; at the same time, $BA[]$ is updated as $BA[i] \leftarrow A[i]$ for all $R_i \in \Delta$.

The tabu list is maintained as follow: When a variable L_x is moved from AVS to WVS, it is inserted at the end of the list. If the tabu list size is bigger than the given tabu tenure, the head of tabu list is deleted. This arrangement makes sure that the tabu list always stores the most updated tabus. When the tabu tenure is large, it may

happen that, in an arbitrary iteration i , all the possible moves are prohibited by the tabu list. This situation will stall the search process since the next current solution cannot be found. To prevent this problem from occurring, we simply delete the first half of the tabu list and re-run iteration i .

The calculation of *CMOVE* for each trial move is described below:

A channel on link i using wavelength w is represented by a duplet (i, w) . We use a two-dimension array $WC[][]$ to store which lightpath occupies a wavelength channel. If a wavelength channel (i, w) is occupied by a lightpath assigned to variable L_x in the current solution, $WC[i][w] = x$; otherwise, $WC[i][w] = Null$. Whenever a lightpath is assigned or deleted, the corresponding element in $WC[][]$ is updated.

For a trial move, after the value is assigned, say $L_x \leftarrow (p_x^i, w)$, we search the assigned lightpath in $WC[][]$. For each $Link_z \in p_x^i$, from the value of $WC[z][w]$, we know which variable in AVS is in conflict with L_x . After all the links belonging to p_x^i are searched, we can get the set of conflicting variables $\{L_y, \dots, L_z\}$. With $\{L_y, \dots, L_z\}$, *CMOVE* can be easily calculated. At the same time, the legitimacy of the trial move is checked by comparing $\{L_y, \dots, L_z\}$ with the current tabu list. If the move is prohibited, a tabu tag is added to the move.

The above-mentioned *CMOVE* calculation only requires $|E| \times |W|$ units of memory and implements τ number of iterations, where τ is the number of hop count of the path p_x^i .

Below are the pseudo codes for the proposed MLE TS algorithm:

```

Begin of main // MLE TS algorithm
Initialize WVS list with  $WVS = \Delta$ ;
Randomly sort WVS list;
 $CAVS \leftarrow 0$ ;  $BCAVS \leftarrow 0$ ;  $cycle \leftarrow 0$ ;
 $A[i] \leftarrow Null$  for all  $R_i \in \Delta$ ;
 $WC[i][w] \leftarrow Null$  for all  $Link_i \in E$  and  $w \in W$ ;
Initialize tabu list as null;
While ( $cycle < CYC$  or  $WVS \neq Null$ ) do //phase 1 begin
Begin
Find the head variables in WVS list;
For each possible value for  $L_x$ , calculate the corresponding  $CMOVE$  and set
tabu tag;
Apply aspiration criteria to tagged moves.
Choose the one with best  $CMOVE$  among all the untagged moves.
 $CAVS \leftarrow CAVS + CMOVE$ ;
Update  $A[]$ ,  $WC[][]$ , WVS list and tabu list according to the accepted move;
If ( $CAVS > BCAVS$ ) then
 $BCAVS \leftarrow CAVS$ ;
 $BA[i] \leftarrow A[i]$  for all the  $R_i \in \Delta$ ;
 $cycle \leftarrow 0$ ;
Else
 $cycle \leftarrow cycle + 1$ ;
Endif
End of While //phase 1 end
 $result \leftarrow Phase2(BA[], BCAVS)$ ;
Output result and  $BA[]$ ;
End of main

```

Phase2 (BA[],BCAVS)

Begin of Subroutine

Initialize $WC[][]$ with $BA[]$;

Initialize WVS list with $BA[]$;

$CAVS \leftarrow BCAVS$; $cycle \leftarrow 0$;

$A[i] \leftarrow BA[i]$ for all the $R_i \in \Delta$;

Initialize tabu list as null;

While ($cycle < CYC$ or $WVS \neq Null$) **do**

Begin

For each variable $L_x \in WVS$ and each possible value of L_x , calculate the corresponding $CMOVE$ and set *tabu* tag ;

Apply *aspiration criteria* to tagged moves.

Choose the one with best $CMOVE$ among all the untagged moves.

$CAVS \leftarrow CAVS + CMOVE$;

Update $A[]$, $WC[][]$, WVS list and tabu list according to the accepted move;

If ($CAVS > BCAVS$) **then**

$BCAVS \leftarrow CAVS$;

$BA[i] \leftarrow A[i]$ for all the $R_i \in \Delta$;

$cycle \leftarrow 0$;

Else

$cycle \leftarrow cycle + 1$;

Endif

End of While

Return $BCAVS$;

End of Subroutine

4.2.6 Simulations

Three network topologies are used in the simulation study (Figure 4.2): NSFNet (14 nodes and 42 unidirectional links)[56], ARPANet (20 nodes and 64 directional links) [50] and BigNet (50 nodes and 164 unidirectional links)[57]. The number of

wavelengths per link is set to 8 for the former two smaller networks and 16 for the large BigNet.

The lightpath request set Δ used in the simulation is randomly generated with at most one request for each node pair.

Figure 4.2 Test topologies

4.2.7.1 Tabu Tenure Selection

As mentioned in [54], *tabu tenure* (tabu list size) is an important parameter, which will significantly affect the performance of a TS algorithm. We study the effect of tabu tenure on our proposed algorithm by running a set of simulations based on NSFNet and BigNet. For NSFNet, four request sets with $|\Delta|=126, 140, 154, 160$ are

tested. For BigNet, five request sets with $|\Delta|=200,300,400,500, 600$ are tested. For NSFNet, the pre-calculated *candidate path set (CPS)* covers all the possible routes for a given request. For BigNet, the cardinality of CPS is set to 5. Five tabu tenure values are tested: 7, $|\Delta|/10$, $|\Delta|/7$, $|\Delta|/5$, $|\Delta|/3$ and $|\Delta|$. Since “7” is a magic number used by many previous applications [54], it is chosen for testing. Other tabu tenure values selected are different fractions of the variable size. Since there are stochastic components in our proposed TS algorithm, for example, in the construction of the WVS list in Phase 1, a different randomly generated seed will lead to a vastly different result. For this reason, ten executions with different random seeds for each test set are run and the best result is chosen. Table 4.1 shows the results for different tabu tenures for NSFNet. Table 4.2 shows the results for different tabu tenures for BigNet. We notice that the tabu tenure of 7 and $|\Delta|/10$ produce more inferior results (shaded entries in Table 4.1 and 4.2) than other tenures that are tested. The results can be explained as follow: When tabu tenure is too small, the tabu list may not contain enough information for the search process to escape from a local optimal solution before the CYC limitation is met. Thus, even for all the ten random starting points, the algorithm may still end at an inferior local optimal. Theoretically, when the tabu tenure is too large, the TS algorithm will search the whole solution space coarsely, since a lot of possible solutions are prohibited by the tabu list, it may lead to inferior results. However, this does not happen in our simulation. This is credited to our method of maintaining the tabu list: When all the possible trial moves are prohibited by the tabu list, entries in the first half of the tabu list are deleted. In other words, the length of the tabu list is dynamically adjusted to the search process even when a large tabu tenure is set. For the case of tabu tenure= $|\Delta|$, the average length of tabu list is

about $|\Delta|/3$ during the simulation. This explains why tabu tenure= $|\Delta|$ yields the same results as tabu tenure= $|\Delta|/3$.

	7	$ \Delta /10$	$ \Delta /7$	$ \Delta /5$	$ \Delta /3$	$ \Delta $
$ \Delta =126$	120	120	120	120	120	120
$ \Delta =140$	124	124	125	125	125	125
$ \Delta =154$	128	129	129	129	129	129
$ \Delta =168$	137	141	141	141	141	141

Table 4.1 Tabu tenure effects for NSFNet

	7	$ \Delta /10$	$ \Delta /7$	$ \Delta /5$	$ \Delta /3$	$ \Delta $
$ \Delta =200$	184	185	182	182	182	182
$ \Delta =300$	252	252	253	253	253	253
$ \Delta =400$	268	275	275	275	275	275
$ \Delta =500$	313	331	332	332	332	332
$ \Delta =600$	346	381	382	382	382	382

Table 4.2 Tabu tenure effects with BigNet

Since tabu tenure = $|\Delta|/7$, $|\Delta|/5$ and $|\Delta|/3$ have almost the same performances (especially for BigNet), we use the middle value $|\Delta|/5$ as tabu tenure for all the following simulations.

4.2.7.2 Consistency of the Proposed Algorithm

Our proposed TS algorithm starts at a random point, which is a result of the program randomly sequencing the WVS list during the initializing stage. Though subsequent search steps are carried out in a deterministic way, different starting points may lead to different end results. To evaluate the consistency of our proposed algorithm, the following simulation is run for the BigNet.

We vary $|\Delta|$ from 200 to 600 with a step increase of 100. Ten request sets are generated for each $|\Delta|$. In total, 50 different Δ 's are tested. Ten executions for each

test set are run with different random seeds. We found that out of the 50 tested request sets, only five request sets yield results. The results for all the ten executions are shown in Table 4.3

Execution Index	Δ =200				Δ =300
	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5
1	185	197	197	191	246
2	185	196	197	191	246
3	185	197	196	190	246
4	185	196	197	190	246
5	185	197	197	191	246
6	184	197	197	190	246
7	184	196	196	191	246
8	185	197	197	191	246
9	183	197	197	189	246
10	185	197	197	188	245

Table 4.3 Number of accepted requests for ten executions

We summarize the average, maximum, and minimum number of accepted requests for each request set in Table 4.4. The maximum difference in the results for different executions is 3 for request set Δ_4 . The maximum ratio of standard deviation to average is 0.54%. This shows that our proposed algorithm has consistent performance.

	Δ =200				Δ =300
	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5
Ave.	184.6	196.7	196.8	190.2	245.9
Max.	185	197	197	191	246
Min.	183	196	196	188	245
Std. Dev. /Ave.	0.38%	0.24%	0.21%	0.54%	0.13%

Table 4.4 Statistics

4.2.7.3 Performance Comparison with ILP

The performance of the proposed TS algorithm is compared with the path-based ILP solution (described in Section 3.1 of this dissertation)[28]. Similar to our TS algorithm, the ILP solution is based on the pre-calculated *CPS* to provide the route for a request. When the pre-calculated *CPS* covers all the possible routes for a given request, the ILP solution is guaranteed to find an optimal solution for the MLE problem. Since the ILP solution is guaranteed to obtain the best result, we use the result obtained by the ILP solution as the benchmark to evaluate the effectiveness of our proposed TS algorithm. The first set of simulations is run for all the three small networks. For NSFnet, we vary $|\Delta|$ from 126 to 168 with a step increase of 14. Ten request sets are generated for each $|\Delta|$. In total, 40 different Δ 's are tested. For a request in Δ , the same *CPS*, which covers all the possible routes for that request, is used in both the TS algorithm and the ILP solution. In this case, the result obtained by the ILP solution is the optimum for the MLE problem. For each test set, the TS algorithm is run independently for five times with different random seeds, and the best result is chosen. The ILP formulation is solved using the ILP problem solver *CPLEX* 8.1. For the 40 request sets tested, the proposed TS algorithm achieves the same result as the ILP solution. For ARPANet, 60 request sets with $|\Delta|$ varying from 120 to 200 with a step increase of 20 are tested. Also, the proposed TS algorithm achieves the same result as the ILP solution. These results are evidence that our proposed TS algorithm is effective in finding optimum solutions for MLE problems for small networks.

	Δ =200		Δ =300		Δ =400		Δ =500		Δ =600	
	ILP	TS	ILP	TS	ILP	TS	ILP	TS	ILP	TS
Δ 1	185	185	253	253	275	275	332	332	382	382
Δ 2	198	198	244	244	287	287	335	335	401	401
Δ 3	192	192	236	236	287	287	342	342	395	395
Δ 4	197	197	236	236	298	298	344	344	375	375
Δ 5	181	181	246	246	276	276	344	344	383	383
Δ 6	196	197	244	244	284	284	336	335	379	379
Δ 7	193	193	250	250	285	285	328	328	390	390
Δ 8	197	197	237	236	292	292	334	334	383	383
Δ 9	189	189	247	247	285	285	329	329	381	381
Δ 10	191	191	250	250	305	305	324	324	372	372

Table 4.5 ILP and TS results for BigNet

Another set of simulation is run for BigNet. We vary $|\Delta|$ from 200 to 600 with a step increase of 100. Ten request sets are generated for each $|\Delta|$. In total, 50 different Δ 's are tested. Due to the much larger computational requirement, we have to limit the cardinality of *CPS* so that the ILP solution can be solved within reasonable time on our desktop computer, which has a 2.4G CPU and 1Gbytes memory. The cardinality of *CPS* is set to 5 in this set of simulation. For each test set, the TS algorithm is run five times independently with different random seeds, and the best result is chosen. The results are shown in Table 4.5. For only two out of the 50 cases (shaded entries in Table 4.5), TS algorithm yields inferior results. The difference is only one request for both cases. The ILP formulation's average solution time for the ten cases with $|\Delta|=600$ is about 8 hours, while our proposed TS algorithm can solve the problem within 10 minutes for five executions.

4.3 MWU Problem with TS Solution

The MWU problem can be formally defined as:

Given a graph $G(E, N)$ and a set of lightpath requests $\Delta = \{R_1, R_2, \dots, R_{|\Delta|}\}$, find a set of wavelengths W , which fulfills the following constraints:

(MUW_C1): For each request $R_i \in \Delta$, a lightpath should be established in the given network.

(MWU_C2): Established lightpaths will strictly follow both the *wavelength continuity constraint* (WCC) and the *lightpath conflict constraint* (LCC).

The objective is to minimize the cardinality $|W|$.

4.3.1 Heuristic Algorithm Using MLE TS Algorithm

We can apply the MLE TS algorithm to solve the MWU in the following way:

The MWU algorithm starts with a relatively large set of available wavelengths W . The initial set of available wavelength set W is constructed using the *first fit* algorithm proposed in [27]. With W and input set Δ , we consider the MLE TS algorithm a subroutine. If the result found by the MLE TS subroutine is equal to $|\Delta|$, it implies that all the requests in Δ can be fulfilled with $|W|$ wavelengths. Then, the cardinality of W is reduced by one, and the MLE TS is run again. The iteration ends when the

solution found by the MLE TS subroutine is smaller than $|\Delta|$ with certain W , and return $|W|+1$ as the result.

The following are the pseudo codes for the MWU algorithm:

Begin of Main // MWU algorithm

Construct the initial W with *first fit* algorithm

$cycle \leftarrow 0$

While { $cycle < 5$ } **do**

Begin

If (MLE_TS(W) = $|\Delta|$) **then**

$|W| \leftarrow |W|-1;$

$cycle \leftarrow 0;$

Else

$cycle \leftarrow cycle + 1;$

Endif

End of While

Return $|W|+1;$

End of Main

As shown in the pseudo codes, before the algorithm stops, we run the MLE TS algorithm five times with different random seeds. From the simulation study of the MLE TS algorithm, it seems quite likely that the MLE TS will find a near-optimal solution with just five executions,

4.3.2 Simulations

The performance of the proposed heuristic algorithm is compared with the ILP solution proposed in [26]. We use the results obtained from the ILP solution as the benchmark to evaluate the effectiveness of our proposed heuristic algorithm. The first

set of simulation is run for NSFNet and ARPANet. The same 40 and 60 test Δ 's in MLE simulations are used. For all the tested Δ 's, the same results are obtained by both the ILP and our heuristic algorithm. It shows that the heuristic is effective in finding optimum solutions for MWU problems for small networks.

Another set of simulation is run for BigNet. The same test Δ 's used in the MLE simulation study, i.e. $|\Delta|$ varies from 200 to 500 with a step increase of 100, are used. The case of $|\Delta|=600$ is not tested since it is too computational intensive for ILP to find a solution. The results show that for all the tested Δ 's with $|\Delta|=200, 300$ and 400 , our TS algorithm yields the same result as the ILP solution. For only 1 out of 10 tested Δ 's of $|\Delta|=500$, the TS algorithm yields an inferior solution (see Figure 4.3a). The difference is one wavelength with the TS solution yielding 47 wavelengths and ILP yielding 46 wavelengths used. The average solution time for the ILP formulation with $|\Delta|=500$ is about 29 hours ($|CPS|=3$) while our proposed TS algorithm can solve the problem within fifteen minutes (see Figure 4.3b).

Figure 4.3 Result for 10 trials with $|\Delta|=500$

4.4 Conclusions and discussion

In this chapter, we first propose a TS algorithm for solving the well-known MLE problem. To the best of our knowledge, we are the first to design a TS algorithm that considers routing and wavelength jointly to achieve greater optimality while keeping computing requirement low. The simulation results obtained confirm that the TS algorithm has excellent performance for both small and large networks. We also propose a heuristic solution based on the proposed MLE TS algorithm for solving the MWU problem. The simulation results show that our heuristic solution works very well in term of optimality and computational cost for the MWU problem.

In this chapter, we only compare our proposed TS search algorithm with the ILP solution. We observe that the algorithm proposed in [29] for solving the MLE problem is able to yield good result for small networks. However, we are unable to use that algorithm on BigNet because LP relaxation is needed for the ILP formulation, which has $O(N(N-1) \times E \times W) = O(50 \times 49 \times 164 \times 8) = 3214400$ variables and is beyond the capability of our software to handle.

We have proved in this chapter that our proposed TS algorithm is effective in solving the SLE problem for large networks. It is interesting to know whether the TS algorithm is suitable also for solving the DLE problem. Note that for the DLE problem, lightpath requests are assumed to arrive at the network in a random manner, and the network needs to decide if it can accommodate a request based on its existing available resource and/or some other operational considerations. The commonly used objective of DLE is to minimize the request-blocking rate [38]. In a scenario in which future incoming requests are not predictable, the goal of minimize the overall

blocking rate is translated into finding a lightpath which utilizes minimum network resource in the absence of other operational considerations. We observe that the most effective algorithm proposed in past studies is the full adaptive algorithm (FAA) [38]. For the FAA, for an optical network with W wavelengths, at most W times shortest path algorithms are needed. However, for the TS algorithm, we need to construct a large feasible solution space in which we search for a near optimal solution. To construct such a solution space, a set of feasible lightpaths need to be found, which is already much more complicated than the FAA. In view of the above, we conjecture that the TS algorithm is not suitable for solving the DLE problem

Chapter 5

Maximum Revenue Problem and Solutions

Despite of the usefulness of the studies on MWU and MLE problems, most network operators are mainly interested in maximizing profit using the existing network facilities. Profit is simply revenue minus cost. Once a network is designed and built, the bulk of the cost has been incurred; thus, it is more important to maximize revenue. We assume that most operators of all-optical networks will derive their revenue from leasing lightpaths to their clients. When fierce competition exists in the market, the price of a lightpath between two locations will be determined more by market forces than by a carrier's true cost. In other words, the price of a given lightpath request is more or less fixed based on the current market situation. In order to maximize revenue, naturally a carrier will prefer to support more lightpaths that command higher premiums. During the lightpath planning stage, an operator is typically faced with two types of lightpath requests. The first type includes those requests that the operator has to fulfill, for example, long-term lease contracts that have to be rolled over or requests that are obliged by provisions mandated by government regulations. The second type includes those requests that the operator has the choice to accept or reject; for example, new lease contracts under negotiation at the planning stage. If there is not enough resource to accommodate all the new contracts, the operator needs to decide which one to accept and which one to reject. In this dissertation, the two categories of requests are organized in two lightpath

request sets; namely, *high priority set* (HPS) and *low priority set* (LPS). As the HPS represents obligation, the operator must accommodate all the requests in the HPS before any request in the LPS can be considered.

The above problem can be formulated as an optimization problem which consists of the constraints that lightpaths in the HPS must be established while the lightpaths in the LPS can be set up selectively to optimize revenue when there is not enough resource to accommodate all. The revenue is basically the sum of the market price of all lightpaths that can be set up. Since we assume that all the requests in the HPS need to be fulfilled, the revenue contribution from these requests is fixed and we only need to maximize the revenue contribution from the LPS demands that can be fulfilled.

5.1 ILP Formulation

The *maximum revenue* (MR) problem can be formally defined as follows:

Given a wavelength routed optical network with $|W|$ number of wavelengths per fiber link, where W is the set of wavelengths a fiber supports, and two lightpath request sets $HPS = \{R_1, R_2, \dots, R_{|HPS|}\}$ and $LPS = \{R_{|HPS|+1}, R_{|HPS|+2}, \dots, R_{|HPS|+|LPS|}\}$.

For each $R_i \in LPS$, a fair market price β_i is given. We need to find a subset

$\Lambda \subseteq LPS$ subject to the following constraints:

(MR_C1): For each request $R_i \in HPS$ or $R_i \in \Lambda$, a lightpath should be established in the given network.

(MR_C2): The established lightpaths will strictly follow both the *wavelength continuity constraint* (WCC) and the *lightpath conflict constraint* (LCC).

The objective is:

$$\text{Maximize } obj_{MR}(\Lambda) = \sum_{R_i \in \Lambda} \beta_i .$$

If a subset $\Lambda \subset LPS$, which fulfills all the constraints, can be found, we call the subset a *feasible subset*. A feasible subset Λ is deemed an *optimal subset* iff $obj_{MR}(\Lambda) \geq obj_{MR}(\Lambda')$ for all the other feasible subset Λ' s.

To find the optimal solution for the MR problem, we propose a path-based ILP formulation. In the path-based formulation, a *candidate path set* (CPS) P_i is pre-calculated for each request R_i . Here, we use p_i^j to denote the j^{th} route in P_i . If P_i includes all the possible routes for R_i on the given graph, the ILP formulation can always find the optimal solution.

Binary Decision Variables:

$L_{i,j}^w : L_{i,j}^w = 1$, if there exists a lightpath using the j^{th} route in P_i and also occupying the w^{th} wavelength; otherwise, $L_{i,j}^w = 0$.

$F_i : F_i = 1$; if a lightpath is set up for request R_i ; otherwise, $F_i = 0$;

ILP Formulation (ILP_MR)

Chapter 5 Maximum Revenue Problems and Solutions

Objective:

$$\text{Maximize } \sum_{R_i \in LPS} F_i \times \beta_i \quad (5-1)$$

Subject to constraints:

$$\sum_{p_i^j \in C_{mn}} L_{i,j}^w \leq 1, \forall (m, n) \in E, w \in W \quad (5-2)$$

$$\sum_{p_i^j \in P_i} \sum_{w \in W} L_{i,j}^w = F_i, \forall (R_i \in (HPS \cup LPS)) \quad (5-3)$$

$$F_i = 1, \forall (R_i \in HPS) \quad (5-4)$$

$$F_i \leq 1, \forall (R_i \in LPS) \quad (5-5)$$

Here, C_{mn} denotes the set of candidate routes, which use link $m \rightarrow n$. C_{mn} can be computed from all the given CPSs.

Equation (5-2) imposes the constraint that no two lightpaths can use the same wavelength on the same link. Equation (5-3) enumerates the number of lightpaths established for each request. Equation (5-4) imposes the constraint that, for each $R_i \in HPS$, there should be an established lightpath. Since all the requests in HPS are assumed to be fulfilled, if there is not enough resource, the ILP formulation would not yield a feasible solution. Equation (5-5) ensures that at most one lightpath is set up for $R_i \in LPS$.

We observe that the number of variables in the ILP grows in $O(\sum_{R_i \in HPS \cup LPS} |P_i| \times |W|)$ and the number of constraints in the ILP grow in $O(|W| \times |E| + 2 \times |HPS \cup LPS|)$. For a large network where $|W|$, $|HPS \cup LPS|$ and $|P_i|$ are large, the ILP problem is virtually unsolvable. In view of this, in the following section, we propose some heuristic algorithms for solving the MR problem.

5.2 Heuristic Algorithms

In this section, two heuristic algorithms are proposed. The first one is a tabu search algorithm, which is based on the TS algorithm for the MLE problem in Chapter 4. A sequential RWA algorithm is also proposed for the purpose of evaluating the proposed TS algorithm for a large network.

5.2.1 TS Algorithm

The TS algorithm proposed in chapter 4 works quite well for MLE problems. There are some similarities between the MLE and MR problems. If we disregard the priority difference between HPS and LPS, and set the fair market price for all requests to unity, the MR problem becomes an MLE problem. This is the basis for adapting the MLE TS algorithm described in Chapter 4 here.

To find a way to drop the priority constraint of the MR problem, we define a *relaxed MR (RMR)* problem as follow:

Given a wavelength routed optical network with $|W|$ number of wavelengths per fiber link, $HPS = \{R_1, R_2, \dots, R_{|HPS|}\}$ and $LPS = \{R_{|HPS|+1}, R_{|HPS|+2}, \dots, R_{|HPS|+|LPS|}\}$,

where W is the set of wavelengths a fiber supports. Associated with each $R_i \in LPS$ is its fair market price of β_i . For each $R_i \in HPS$, the same artificial market price of β_{HPS} is assigned. We need to find a subset $\Lambda \subseteq (HPS \cup LPS)$ subject to the following constraints:

(RMR_C1): For each request $R_i \in \Lambda$, a lightpath should be established in the given network.

(RMR_C2): The established lightpaths will strictly follow both the *wavelength continuity constraint* (WCC) and the *lightpath conflict constraint* (LCC).

The objective is:

$$\text{Maximize } obj_{RMR}(\Lambda) = \sum_{R_i \in \Lambda} \beta_i$$

If a subset $\Lambda \subseteq (HPS \cup LPS)$ that fulfills all the constraints can be found, we call the subset a *feasible subset*. A feasible subset Λ is an *optimal subset* iff $obj_{RMR}(\Lambda) \geq obj_{RMR}(\Lambda')$ for all the other feasible subset Λ' 's.

In the RMR problem formulation, the priority constraint previously exists in the MR problem formulation, is dropped. Basically that priority constraint in the MR problem requires every request in the *high priority set* (HPS) to be fulfilled. Hence, for the RMR problem, an optimal request set Λ found does not need to contain all the requests in HPS. However, in what follows, we will prove that when β_{HPS} is

sufficiently large; the optimal subset for the RMR problem is also the optimal subset for the MR problem.

Theorem 5.1: When $\beta_{HPS} > \sum_{R_i \in LPS} \beta_i$ and there is an optimal subset $\Lambda \subseteq (HPS \cup LPS)$ for the RMR problem; if $obj_{RMR}(\Lambda) \geq |HPS| \times \beta_{HPS}$, then Λ can be represented as $\Lambda = HPS \cup \Lambda'$, where $\Lambda' \subseteq LPS$, and Λ' is also an optimal subset for the MR problem.

Proof:

An optimal set for the RMR problem can be represented as: $\Lambda = \Psi \cup \Lambda'$, where $\Psi \subseteq HPS$ and $\Lambda' \subseteq LPS$. $obj_{RMR}(\Lambda)$ can be computed from:

$$obj_{RMR}(\Lambda) = \sum_{R_i \in (\Psi \cup \Lambda')} \beta_i = |\Psi| \times \beta_{HPS} + \sum_{R_i \in \Lambda'} \beta_i.$$

With the assumption of $obj_{RMR}(\Lambda) \geq |HPS| \times \beta_{HPS}$, we have:

$$|\Psi| \times \beta_{HPS} + \sum_{R_i \in \Lambda'} \beta_i \geq |HPS| \times \beta_{HPS} \Rightarrow \sum_{R_i \in \Lambda'} \beta_i \geq |HPS| \times \beta_{HPS} - |\Psi| \times \beta_{HPS}$$

Since we also assume that $\beta_{HPS} > \sum_{R_i \in LPS} \beta_i$, then only when $|\Psi| = |HPS|$, the above inequality holds. In other words, $\Psi = HPS$.

So far, we have proved that $\Lambda = HPS \cup \Lambda'$. Since Λ meet all the constraints of RMR, we know for sure that all the requests in HPS and all the requests that satisfy $R_i \in \Lambda'$ can be fulfilled at the same time for the given network. From the definition of

MR, we know $\Lambda' \subseteq LPS$ is also a feasible set for MR. Now, we can proceed to prove that Λ' is an optimal set for the MR problem, or $obj_{MR}(\Lambda') \geq obj_{MR}(\Lambda'')$ for any other feasible set Λ'' for the MR problem.

From the formulation of the MR problem, a set Λ'' is deemed to be feasible if all the requests in HPS and all the requests that satisfy $R_i \in \Lambda''$ can be fulfilled at the same time by the given network. Thus $HPS \cup \Lambda''$ is also a feasible set for the RMR problem. Since we assume that Λ is an optimal set for the RMR problem, we have

$$obj_{RMR}(\Lambda) \geq obj_{RMR}(HPS \cup \Lambda'') \Rightarrow$$

$$|HPS| \times \beta_{HPS} + \sum_{R_i \in \Lambda'} \beta_i \geq |HPS| \times \beta_{HPS} + \sum_{R_i \in \Lambda''} \beta_i \Rightarrow \sum_{R_i \in \Lambda'} \beta_i \geq \sum_{R_i \in \Lambda''} \beta_i$$

$$\Rightarrow obj_{MR}(\Lambda') \geq obj_{MR}(\Lambda'')$$

Q.E.D

Theorem 5.1 suggests that if we set $\beta_{HPS} > \sum_{R_i \in LPS} \beta_i$, solving the RMR problem yields the optimal subsets for the MR problem. In the following, we will show that solving the MR problem yields the optimal subsets for the RMR problem too.

Theorem 5.2: if $\beta_{HPS} > \sum_{R_i \in LPS} \beta_i$ and there is an optimal subset $\Lambda \subseteq LPS$ for the

MR problem, then, $\Lambda \cup HPM$ is also an optimal subset for the RMR problem.

Proof:

From the assumption that an optimal set Λ exists for the MR problem, we know that all the requests in HPS and all the requests that satisfy $R_i \in \Lambda$ can be fulfilled at the same time by the given network; for all the other feasible subsets $\Lambda' \subseteq LPS$, $obj_{MR}(\Lambda) \geq obj_{MR}(\Lambda')$. Clearly, the set $\Lambda \cup HPS$ is a feasible set for the RMR problem. Now, we need to prove that $obj_{RMR}(\Lambda \cup HPS) \geq obj_{RMR}(\Lambda'')$ for all the other feasible subsets $\Lambda'' \subseteq (HPS \cup LPS)$. First, we can find $obj_{RMR}(\Lambda \cup HPS)$ using the following equation:

$$\begin{aligned} obj_{RMR}(\Lambda \cup HPS) &= \sum_{R_i \in (\Lambda \cup HPS)} \beta_i = |HPS| \times \beta_{HPS} + \sum_{R_i \in \Lambda} \beta_i \\ &= |HPS| \times \beta_{HPS} + obj_{MR}(\Lambda) \end{aligned}$$

If $obj_{RMR}(\Lambda'') < |HPS| \times \beta_{HPS}$, clearly $obj_{RMR}(\Lambda \cup HPS) > obj_{RMR}(\Lambda'')$.

If $obj_{RMR}(\Lambda'') \geq |HPS| \times \beta_{HPS}$, from the proof of Theorem 5.1, we know that $\Lambda'' = HPS \cup \Lambda'$, and Λ' is a feasible set for the MR problem. Then

$$obj_{RMR}(\Lambda'') = |HPS| \times \beta_{HPS} + obj_{MR}(\Lambda')$$

$$obj_{RMR}(\Lambda \cup HPS) - obj_{RMR}(\Lambda'') = obj_{MR}(\Lambda) - obj_{MR}(\Lambda') \geq 0$$

Q.E.D

With Theorem 5.1 and 5.2, we know that solving the RMR problem is equivalent to solving the MR problem. The difference between the two problems is that, in MLE, the objective is to maximize $|\Lambda|$; and in RMR, the objective is to maximize $\sum_{R_i \in \Lambda} \beta_i$.

In the MLE TS algorithm, we use COST function to evaluate the fitness of a feasible solution. The change in objective can be reflected by changing the COST function. In the MLE TS algorithm, we use COST function $CAVS$ to evaluate the current solution and $COMVE$ to evaluate a trial move. $CAVS$ is calculated as the number of variables in AVS and $CMOVE$ is calculated as the cardinality change of AVS caused by the trial move. Now, in the RMR TS algorithm, we only need to modify the $CAVS$ and $CMOVE$ functions of the MLE TS algorithm. Since the objective of RMR is to maximize the revenue of the accepted lightpaths, we modify $CAVS$ as the revenue generated by the accepted requests with $CAVS = \sum_{R_i \in AVS} \beta_i$. Then, a bigger $CAVS$ means more revenue is generated for a given network. $COMVE$ is calculated as the revenue change caused by a trial change with $CMOVE = \beta_x - \sum_{L_y \in \Phi} \beta_y$, where β_x is the market price for a newly added variable L_x in AVS and Φ is the set of variables deleted from AVS.

5.2.2 Sequential RWA Algorithm

A heuristic *sequential RWA (SR)* algorithm is devised to mimic the algorithms used in previous studies [37][39] to solve the MLE problem. The SR algorithm works as follow: In Step 1, the requests in the HPS and LPS are sorted separately based on certain criteria. For each request in the HPS, the shortest path is found in the given graph $G(N, E)$. Then, the requests are sorted in descending order according to their shortest path lengths to form a request list A . The requests in the LPS are sorted in descending order according to their market prices to form List B . In Step 2, the route and wavelength for the first request in List A is determined by means of the full-

adaptive algorithm described in [39]. After the first lightpath is found, the resource used by that lightpath is removed from the network resource availability data. Then, the process is repeated for the next request in List A . If the algorithm fails in finding a lightpath for a request in List A due to lack of resources, the algorithm stops, which means the algorithm has failed to solve the MR problem because we need all the requests in the HPS to be met. After all the requests in List A are fulfilled, the algorithm begins to find a lightpath for the first request in List B . If it succeeds, the network resource availability data is updated accordingly; if not, the request is skipped and the next request is considered. The process is repeated until all the requests in List B have been considered.

5.3 Simulations

5.3.1 Performance Comparison for Small Networks

In the simulation, the performance of the ILP, TS and SR algorithms are compared. For the first set of simulations, we use the two small networks described in Chapter 4; they are, namely, NSFNet (14 nodes and 42 unidirectional links) and ARPANet (20 nodes and 64 directional links). The number of wavelengths per link is set to 8. The HPS and LPS are randomly generated with at most one request for each node pair. For NSFNet, $|HPS|$ varies from 42 to 84 with a step increase of 14. Five request sets are generated for each $|HPS|$ and in total 20 HPSs are tested. For each HPS tested, a LPS with $|LPS| = |N| \times (|N| - 1) / 2 = 91$ is randomly generated, where $|N|$ is the number of nodes in the network. For ARPANet, $|HPS|$ varies from 20 to 80 with a

step increase of 20. Five request sets are generated for each $|HPS|$ and in total 20 HPSs are tested. For each HPS tested, an LPS with $|LPS| = 20 \times (20 - 1) / 2 = 190$ is randomly generated.

We consider two market price models. The first model is dubbed *proportioned price*, for which each request in LPS is assigned a price that is equal to the hop-count of the shortest path for that request. This reflects the market situation in which the price of a lightpath is proportioned to a carrier's own cost. The second model is dubbed *random price*, where each request in LPS is assigned a price drawn from a uniform distribution. This assignment mimics the market situation in which the price of a lightpath is no longer decided by a carrier's own cost but by market forces.

The *candidate paths set* (CPS) in this set of simulations contains all the possible routes for a request, thus the result yielded by the ILP is optimal. The ILP formulation is solved using the ILP problem solver *CPLEX* 8.1. For the TS algorithm, ten executions are carried out and the best result is chosen. The tabu tenure is $(|HPS| + |LPS|) / 5$, the same as the one we used in the MLE TS algorithm.

Simulation results are plotted as revenue versus $|HPS|$ in Figure 5.1(for NSFNet) and Figure 5.2(for ARPANet). The revenue generated by the five tested LPS sets with the same $|HPS|$ are averaged. Note that when the number of requests in the HPS increases, there is less capacity left for the requests in the LPS, thus the revenue contributed by the requests in the LPS decreases. As shown in Figures 5.1 and 5.2, for both market price models, the performance of TS is much better than SR and only slightly inferior to the optimal revenue obtained by solving the ILP formulation.

a) Revenue vs. |HPS|, Proportioned Price Model

b) Revenue vs. |HPS|, Random Price Model

Figure 5.1 Results for NSFNet

a) Revenue vs. |HPS|, Proportioned Price

b) Revenue vs. |HPS|, Random Price

Figure 5.2 Results for ARPANet

Table 5.1 and 5.2 compare the normalized performance differences between different algorithms. As can be seen, when the amount of traffic in the HPS increases, the performance of SR deteriorates rapidly while that of the TS algorithm remains close to the optimal one. For SR, the route and the wavelength for a request are computed based on the state of the network at the moment the request is processed, which may be sub-optimal from the global perspective. When there are enough resources, the sub-optimality of a previously created lightpath may have little impact on the resource availability for new requests. However, when spare resources become scarce, the sub-optimality of a previously created lightpath will make it difficult for resources to be found for new requests.

	(ILP-TS)/ILP	(ILP-SR)/ILP
HPS =42	1.1%	11.0%
HPS =56	1.8%	12.3%
HPS =70	3.2%	14.8%
HPS =84	4.1%	21.9%

a) Performance comparison, Proportioned Price

	(ILP-TS)/ILP	(ILP-SR)/ILP
HPS =42	1.1%	7.9%
HPS =56	0.7%	7.6%
HPS =70	2.1%	13.0%
HPS =84	3.5%	17.0%

b) Performance comparison, Random Price

Table 5.1 Results for NSFNet

	(ILP-TS)/ILP	(ILP-SR)/ILP
$ HPS =20$	2.7%	6.2%
$ HPS =40$	3.3%	11.8%
$ HPS =60$	4.9%	13.0%
$ HPS =80$	6.1%	15.4%

a) Performance comparison, Proportioned Price

	(ILP-TS)/ILP	(ILP-SR)/ILP
$ HPS =20$	0.5%	5.2%
$ HPS =40$	1.3%	7.5%
$ HPS =60$	2.8%	11.3%
$ HPS =80$	4.2%	13.3%

b) Performance comparison, Random Price

Table 5.2 Results for ARPANet

5.3.2 Performance Comparison for a Large Network

The second set of results is obtained from the large network BigNet (50 nodes and 164 unidirectional links, Chapter 4). The number of wavelength channels per link is set to $|W|=20$. $|HPS|$ varies from 50 to 200 with a step increase of 50. Five request sets are generated for each $|HPS|$, and in total 20 HPSs are tested. For each HPS tested, an LPS with $|LPS|=200$ is randomly generated. Due to the much larger computational requirement, we have to limit the cardinality of CPS so that the ILP solution can be solved within reasonable time on our desktop computer with 2.4G CPU and 1Gbytes memory. The cardinality of CPS is set to 5 in this set of simulation.

Figure 5.3 shows that, for this large network, the TS algorithm continue to outperform the SR algorithm and is only slightly inferior to the ILP solution. The

largest performance difference between ILP and TS is less than 5% for both price models. The largest performance difference between ILP and SR is around 15%.

a) Revenue vs. |HPS|, Proportioned Price

b) Revenue vs. |HPS|, Random Price

Figure 5.3 Results for BigNet

The SR algorithm found the solution within 10 seconds for all the cases tested. The TS algorithm used several minutes to find a solution. The ILP method consistently took much longer time. Figure 5.4 shows the average solution time for the TS and ILP approaches.

Figure 5.4 Running time comparison

5.4 Conclusions and discussions

In this chapter, the problem of maximizing the revenue of an optical network with lightpaths having two different priority levels is studied. We develop three solutions: ILP, a heuristic solution based on the MLE TS algorithm, and a *sequential* RWA

algorithm (SR) that mimics those algorithms commonly used for solving the MLE problem. The ILP approach is only suitable for small networks. The proposed TS algorithm is much more scalable compared to ILP, and works for large networks. The SR algorithm is most computationally efficient. In term of the optimality of solution found, TS is only slightly inferior to ILP and much better than SR. To conclude, TS is the best compromise for the maximum revenue problem studied in this chapter.

In our study, we assume that once a network is designed and built, the bulk of the cost has been incurred; thus, it is more important to maximize revenue. In some cases, the operational cost may also be a consideration for network operators. For example, the cost of setting up a lightpath may vary when different physical facilities need to be used. This type of consideration can be satisfied by a slight modification of the cost function in our TS algorithm. In our TS algorithm, we use a set of candidate lightpaths CPS_i for each request R_i . We can assign a cost $CL_{i,j}$ to each candidate lightpath $L_{i,j} \in CPS_i$, where j is the index of the lightpath in CPS_i , to reflect the operational cost. Then, the cost function CAVS for evaluating the fitness of the accepted requests can be changed to $CAVS = \sum_{L_{i,j} \in AVS} (\beta_i - CL_{i,j})$ to incorporate the operating cost consideration.

Chapter 6

Enquiry Process for IP-over-optical Networks Based on the Overlay Model

For an IP-over-optical network, three interconnection models between the control planes of the IP layer and the optical layer are possible; namely, overlay, augmented (also known as inter-domain), and peer models [4]. The overlay model is a hierarchical structure where the IP layer resides on top of the optical layer in a client-server fashion. Each layer has its own routing protocol and has no knowledge of the resource and topology information of the other layer. When an LSP request reaches an LSR, it can either be routed in the IP domain or spawn a lightpath request to the lower optical layer via a *user-to-network interface* (UNI) between the two layers. Exactly how a connection request is handled in each layer is totally decided by the layer itself and will not interfere with the other layers.

For the augmented model, while each layer may run its own routing protocol independently, selected routing information can be exchanged across the UNI. This is analogous to how a border router re-distributes routing information between two IP sub-networks that may be running two different routing protocols.

For the peer model, the control plane of the optical layer is regarded as a peer of the IP/MPLS layer. In this case, from the control plane point of view, an OXC is

regarded as another LSR in the network. A routing protocol such as OSPF or IS-IS runs over both the IP/MPLS and optical layer. The topology and link state information maintained by an LSR and an OXC can (but do not necessarily have to) be the same. This flexibility allows an LSR to calculate an end-to-end path across the whole network, e.g., traversing a mixture of LSRs and OXCs, purely through LSRs or purely through OXCs. Compared with the overlay model, the peer model has the advantage of using global information (of both the IP and optical layers) to optimize IP traffic management and optical resource control. This makes integrated routing possible for the peer model by taking the spare capacity in both layers into consideration.

Figure 6.1 Routing comparison between overlay model and peer model

The difference in routing for overlay model and peer model can be explained by means of Figure 6.1. In Figure 6.1 a), a snapshot of the network state is shown. In the figure, the broken lines (marked with X in the graph) represent the existing lightpaths in the optical layer. Figure 6.1 b) shows the different routing topologies as seen by the

IP layer and optical layer under the overlay model. Figure 6.1 c) shows the integrated routing topology as seen by the control entity under the peer model. The dotted lines in Figures 6.1 b) and 6.1 c) represent the virtual links (existing lightpath) between LSRs. In Figure 6.1 c), the link between an OXC and an LSR represents the add-drop ports of an OXC. Consider the situation in which there is a request to set up an LSP between LSR C and B . For the overlay model, there are two paths in the directed IP layer graph; namely, $C \rightarrow B$ and $C \rightarrow D \rightarrow B$. If there is insufficient residue bandwidth in virtual link $C \rightarrow B$, and virtual link $C \rightarrow D$, the LSP cannot be established in the IP layer. If there is also insufficient resource, say on link D to B and link A to B , to create a lightpath between OXC C and B , the LSP request has to be rejected under the overlay model. However, for integrated routing under the peer model, as long as there is enough residue bandwidth in virtual link $D \rightarrow B$ in the IP layer and a lightpath between C and D can be created in the optical layer, the LSP request can still be accommodated.

In most cases, IP networks and *optical transport networks* (OTNs) are often owned by different administrations. It is highly unlikely that the operator who owns an OTN will allow its upper-layer clients to have the control of its resource. This makes the peer model impractical despite its obvious performance advantages [4][52]. In addition, the peer model approach requires that routing information specific to optical networks be made known to IP routers; this has the tendency to overburden the IP routers, which are typically required to respond quickly to routing requests in the IP layer.

From resource utilization point of view, the key factor that results in the inferior performance of the overlay model is that the IP layer lacks the resource information of

the optical layer. In view of this, we propose a novel enquiry mechanism that can be invoked during the signaling process of LSP provisioning. The enquiry process will help the IP layer to obtain certain resource information from the optical layer. With the information, the IP layer can make an informed decision in choosing an appropriate route for an LSP request to better utilize the available network resource.

6.1 Proposed Enquiry Mechanism

Based on the description in [4], in what follows, we summarize the operational processes of the overlay model.

The optical layer allows an edge LSR to register and query for external addresses. For example, the edge LSR summarizes the IP sub-networks to which it is connected, and registers the sub-network prefixes along with its own address to the OXC to which it is connected via a UNI. After accepting the registration, the corresponding OXC disseminates the routing information as an *autonomous system* (AS) external route to other OXCs via the OSPF routing protocol running in the optical layer. This arrangement ensures that each OXC knows which external IP address is reachable, and via which OXC. When an edge LSR issues a lightpath request message via its UNI, the message will include the addresses of the source and destination LSRs. Then, the corresponding XC can map the LSR addresses to the OXC address so that routing and wavelength assignment for the request can be carried out. An established lightpath is registered as a *forward adjacency* (FA) in the IP layer, which is essentially a *virtual link* that can be advertised by a link-state protocol. This collection

of established lightpaths defines the virtual topology of the IP layer. OSPF is run on this virtual topology to exchange routing information for the IP layer.

Under the conventional overlay model-based service provisioning, two types of service provisioning strategies may be adopted, i.e., IP-layer-first and optical-layer-first [50]. For the first strategy, the LSP request is first routed in the IP layer; only if the IP layer does not have enough capacity to accommodate it will the request be directed to the optical layer. The optical layer will try to establish a lightpath between the corresponding OXC node pair. If the attempt is successful, it will establish the lightpath and use it to accommodate the LSP. At the same time, the IP layer will update its link state database to include this newly established virtual link (i.e., lightpath). If the attempt fails, the LSP request will be blocked. In contrast, for the optical-layer-first strategy, the LSP request will always result in the creation of a new lightpath in the optical layer. If the lightpath can be established successfully, it is used to serve the LSP and the link state database of the IP layer will be updated; otherwise, the request is directed to the IP layer. If both the layers are unable to fulfill the request, the request is blocked.

Note that in both these approaches, the spare resource of only one layer (either the IP layer or the optical layer) may be utilized in accommodating the request and that the resources of both layers cannot be combined. This is inefficient in a number of scenarios like the one illustrated in the following example: Assume that a request to set up an LSP between LSR 1 and 6 of a network for a specific bandwidth requirement has just arrived at LSR 1 (refer to Figure 6.2). Further assume that in the view of source node LSR 1, the IP layer topology is as depicted in the top diagram of Figure 6.2 after we remove all those virtual links that cannot accommodate the request.

If the resource is inadequate to establish a new lightpath from OXC 1 to 6 in the optical layer, the LSP request would be rejected by the standard IP-layer-first or optical-layer-first provisioning scheme. However, if the IP layer can obtain some resource information from the optical layer, it may be possible for the request to be routed by utilizing the spare resources of both layers. For example, in Figure 6.2, the IP layer can send *lightpath enquiries* to the optical layer to check the availability of a new lightpath between OXCs 2 and 6, and/or between OXCs 9 and 6. If any of the enquiries results in a positive response, the routing request may be satisfied by using (existing virtual link $L_1 \rightarrow L_2$, new virtual link $L_2 \rightarrow L_6$) or (existing virtual link $L_1 \rightarrow L_9$, new virtual link $L_9 \rightarrow L_6$).

Figure 6.2 Example of IP-over-Optical network

If an enquiry for a new virtual link from LSR_x to LSR_y is deemed necessary by the ingress LSP, an appropriate message with the addresses of LSR_x and LSR_y is forwarded to the optical layer. The optical layer knows which LSR is reachable through which OXC and this enables the optical layer to find out the corresponding OXC addresses. When the source and destination OXC addresses are known, the optical layer can determine if there is enough wavelength resource for a lightpath between the two OXCs. If the optical layer succeeds in finding such resource, it will return a positive reply to the IP layer containing the cost of that lightpath; otherwise, a negative reply is returned. Note that the enquiry process does not undermine the independence of the optical layer control plane because, based on its own control policy, the optical layer has the liberty to return a negative response even when optical resource for the enquiry is indeed available.

6.1.1 Enquiry Process

Based on the routing algorithm chosen by the IP layer, the ingress LSR LSR_i decides which virtual link (lightpath) needs to be queried, say link $LSR_x \rightarrow LSR_y$.

Two possible ways to implement the enquiry process are as follows:

Figure 6.3 Enquiry process scheme A

Scheme A: LSR_i sends a *virtual link enquiry message* (VLEM) with LSR_y 's address to LSR_x , via the control plane of the IP layer. After receiving the enquiry message, LSR_x forwards it to OXC_x , to which it is connected via a UNI. Then, OXC_x carries out optical layer address mapping and performs the necessary processing for the queried lightpath. The query result is sent back from OXC_x to LSR_x , and then to LSR_i via an *enquiry response message* (ERM).

Scheme B: LSR_i sends a *VLEM* that contains the addresses of LSR_x and LSR_y to OXC_i , to which it is connected via a UNI. After that, OXC_i carries out the address mapping operation. In this scheme, OXC_i can either compute the enquired lightpath by itself or direct the enquiry to the source end of the enquired lightpath, i.e., OXC_x . Which OXC is responsible for carrying out the necessary processing for the enquired lightpath depends on the optical layer's own control policy. For example, the optical layer may choose a fully adaptive RWA algorithm [39], in which each OXC in the optical layer is assumed to maintain up-to-date network state and can carry out the necessary processing for a lightpath that is not originated from itself. In this case, OXC_i can carry out the necessary processing for the queried lightpath all by itself. Alternatively, the optical layer may choose a fixed-alternate RWA scheme [42]. For such a scheme, each OXC stores a routing table that contains an ordered list of a limited number of predefined routes for each destination OXC. An OXC can process a queried lightpath originating from itself by checking the list of predetermined routes in sequence until a route with a valid wavelength assignment is found. In the fixed-

alternate RWA scheme, OXC_i cannot process a lightpath that is originated from other OXCs. In this situation, OXC_i has to forward the enquiry message to OXC_x in the enquiry process.

Figure 6.4 Enquiry process scheme B

Clearly, Scheme A will add more signaling overhead to the IP layer control plane while scheme B will generate more signaling overhead to the optical layer control plane. Which scheme to use for the provisioning of a specific LSP request can be flexibly determined by the ingress LSR so that the signaling overhead can be appropriately distributed to both layers. Scheme B is preferred when the optical layer uses a fully adaptive RWA scheme because signaling messages for lightpath enquiry are only exchanged via the UNI; there is no additional signaling required within the IP layer and optical layer control planes.

6.1.2 Enquiry Message Format

Two types of enquiry messages are used in the enquiry process, namely, *virtual link enquiry message* and *enquiry response message*.

In Scheme A, the format of the *virtual link enquiry message* may take the following form (VLEM format1):

```
Host_LSR_Addr, Res_LSR_Addr,
LSP_Id, Num_VL,
Des_LSR_Addr,
.
.
.
Des_LSR_Addr
```

In this format, Host_LSR_Addr contains the address of the host LSR that initiated the enquiry process. Res_LSR_Addr contains the address of the LSR that is responsible for sending an enquiry to the optical layer via the UNI. Res_LSR_Addr is used by the network to route the VLEM through the IP control plane while Host_LSR_Addr is used to inform the responding LSR where to send back the enquiry result.

Since multiple LSP requests may reach the host LSR within a short time period, the stochastic nature of the processing and transmission delay of enquiry messages may lead to some confusion among different enquiry processes. For this reason, there is a need for the inclusion of LSP_Id to keep track of message sequences.

The enquiry process may involve only a single new virtual link or a set of new virtual links, depending on the routing algorithm chosen by the IP layer. Thus,

Num_VL is needed to indicate how many new virtual links, which are originated from the same LSR but have different destination LSRs, need to be queried. Clearly, using one (albeit large) enquiry message to query a set of lightpaths will be more efficient than sending separate enquiry messages for each virtual link.

For Scheme A, the ingress LSR_i will assign an LSP id to each incoming request. The LSP id is maintained locally by the LSR_i and is contained in the LSP_Id field of VLEM. LSR_i will fill in its own address in the Host_LSR_Addr field. If virtual links $LSR_x \rightarrow LSR_y$ and $LSR_x \rightarrow LSR_z$ are the subjects of the enquiry, LSR_i will fill in the address of LSR_x in Res_LSR_Addr, and set the NUM_VL field to 2. The addresses of LSR_y and LSR_z are carried in the Des_LSR_Addr field in sequence.

For Scheme A, the format of the *enquiry response message* is similar to the *virtual link enquiry request message*, but with some additional data fields (ERM format1):

```

Res_LSR_Addr, Host_LSR_Addr,
LSP_Id, Num_VL,
Des_LSR_Addr, Cost
.
.
.
Des_LSR_Addr, Cost

```

After receiving the VLEM, OXC_x will map the Des_LSR_Addr addresses into OXC addresses and process the information needed for all the enquired lightpaths. The various Cost fields in the *enquiry response message* are set to reflect the ‘cost’ of the corresponding lightpaths. The ‘cost’ may be computed in a way to reflect the propagation delay, leasing cost, hop count of a lightpath or other measures, depending

on the operator's operating concerns. The cost field may even be set to infinity to indicate that an enquired lightpath is not available. This may be due to lack of adequate resources for this enquired lightpath enquiry or some other reasons.

For Scheme B, the format of the *virtual link enquiry message* may take the following form (VLEM format2):

```
Host_LSR_Addr
LSP_Id, Num_VL,
Src_LSR_Addr,Des_LSR_Addr,
.
.
.
Src_LSR_Addr,Des_LSR_Addr,
```

Since there may be more than one LSR connected to the same OXC, Host_LSR_Addr is used to inform the OXC which LSR initiated this request. Src_LSR_Addr and Des_LSR_Addr contain the source and destination LSR addresses of the enquired virtual links. LSR_i sends the VLEM through the UNI to OXC_i . After the OXC_i has carried out the address mapping, it can either process the necessary information for the enquired lightpath by itself or break this enquiry into a number of individual lightpath enquiries, and then forward the individual enquiries to the OXCs concerned.

The format of the *virtual link enquiry message* in the optical layer may take the following form (VLEM format3):

Host_OXC_Addr, Res_OXC_Addr,
 Enquiry_Id, Num_VL,
 Des_OXC_Addr,
 .
 .
 .
 Des_OXC_Addr

OXC_i will fill in the Host_OXC_Addr field with its own address and the Res_LSR_Addr and Des_OXC_Addr fields with the corresponding OXC addresses. To identify a specific enquiry request from a specific LSR to which it is connected to, OXC_i can use the Host_LSR_address and LSP_Id of the VLEM (in format2) it receives from the UNI to generate an Enquiry_Id, and maintains the Enquiry_Id locally.

The format of the *enquiry response message* in the optical layer may take the follow form (ERM format3):

Host_OXC_Addr, Res_OXC_Addr,
 Enquiry_Id, Num_VL,
 Des_OXC_Addr, Cost
 .
 .
 .
 Des_OXC_Addr, Cost

The format of the *enquiry response message* sent back by OXC_i to LSR_i via the UNI may take the following form (ERM format2):

Host_LSR_Addr
LSP_Id, Num_VL,
Src_LSR_Addr,Des_LSR_Addr, Cost
.
.
.
Src_LSR_Addr,Des_LSR_Addr, Cost

6.1.3 Resource Allocation

The goal of the enquiry process is simply to give the IP layer a way to find out if certain spare capacity is available in the optical layer. How to decide which lightpaths need to be enquired, and whether or not the enquired lightpaths will actually be used will be solely governed by the IP layer routing algorithm.

Resource reservation or allocation for the enquired lightpath is not required at the enquiry stage.

6.3 Provisioning Scheme with Enquiry Process

When an LSP request reaches the IP/MPLS domain, the ingress LSR that makes the *explicit routing* [11] decision will decide if any new virtual links should be enquired based on the routing topology it maintains and routing algorithm it chooses. The enquiry process is invoked if certain new virtual links need to be enquired. Based on the enquiry result at the end of the enquiry process, the ingress LSR may reject the LSP request or may serve the request by requesting the optical layer to create one or more new lightpaths. Note that at the enquiry stage, the creation of a lightpath that is

being queried is not required since the ingress LSR may choose not to use the queried lightpath in the final route that it chooses.

Once the new lightpaths are established, the link states of both layers should be updated. The optical layer will update the available wavelength resources on each link while the IP layer will add the established lightpaths as new virtual links in its topology map. The IP layer will use the new topology map to serve the LSP request. After the LSP is established, the link states in the IP layer will also be updated.

The LSP release process of the proposed scheme is similar to that of a conventional overlay model-based provisioning method. Upon receiving the release request, the system will release all the resources used by the LSP. Moreover, if this results in some lightpaths becoming totally unused, these lightpaths will be released. After each release, the network link state information of the affected layers will be updated accordingly.

6.4 Conclusions

We have described an enquiry process which can be used by the IP layer to obtain from the optical layer information on the availability of certain capacity, which is useful for the provisioning of an LSP request. The proposed enquiry process does not violate the layer independency feature of the overlay model but has the advantage of having useful availability information on optical layer resource to allow for better resource allocation and utilization.

The enquiry process can significantly improve network resource utilization and decrease the LSP request blocking ratio in, but not limited to, the following situations:

- a) The LSP request cannot be routed solely in the IP layer based on the existing IP layer topology and resource. In this case, the ingress LSR can find out if there is spare resource in the optical layer for the LSP request to be fulfilled.
- b) The LSP request can be routed solely in the IP layer; however, it may be possible for the enquiry process to find a more capacity efficient route with spare resource in the optical layer.

The decision with regard to which new virtual links should be enquired is governed by the routing algorithm chosen by the IP layer. Since each lightpath enquiry will result in some processing in the optical layer, the routing algorithm in the IP layer should be carefully designed so that the enquiry process does not incur too much signaling and control overhead. In the following chapter, we will describe two simple routing algorithms which can make use of the enquiry process to find LSP routes with near-optimal resource usage.

Chapter 7

Routing Algorithms for Enquiry-based Provisioning Scheme

Figure 7.1 Example for an enquiry-based routing algorithm

While the enquiry process described in Chapter 6 provides a mechanism for the IP layer to obtain certain resource information from the optical layer, we have to overcome another problem when we try to route LSP requests to achieve optimal resource usage. We will make use of the example illustrated in Figure 7.1 to exemplify the problem. In Figure 7.1 a), a four-node IP layer with the existing virtual links is shown. The number shown beside each link is the cost of that link. To reflect the optical layer resource usage of a virtual link, we define the cost of a virtual link to be the number of wavelength channels traversed in the optical layer. We assume there is an LSP request from LSR_1 to LSR_3 , and that the existing virtual links have enough

remaining bandwidth for that request. To find an optimal resource usage route, the ingress LSR needs to find all the possible routes for that request and then chooses the minimum cost one. In this example, with the help of the enquiry process, LSR_I found five possible candidate routes: (VL_{13}) , (VL_{12}, VL_{23}) , (VL_{14}, VL_{43}) , $(VL_{12}, VL_{24}, VL_{43})$ and $(VL_{14}, VL_{42}, VL_{23})$, and $(VL_{12}, VL_{24}, VL_{43})$ turns out to be the optimal route. To find all these routes, we need to make enquiries for all of the five new virtual links which are represented as broken lines in Figure 7.1 b). This approach of finding an optimal route may be implemented as a two-step process. First, the ingress LSR builds a final routing topology (Figure 7.1 b in the example) with the help of the enquiry process and then solves the shortest path problem of the final routing topology. Note that in this first step, the ingress LSR does not know which new virtual links belong to the optimal route. In order to build a final routing topology, enquiries need to be made for all the “possible new links” to ensure that the optimal route will definitely be included at this stage. (If for some reason, this optimal route is not included at this stage, subsequent steps will not be able to find this optimal route.) Our approach requires that if there is no existing virtual link between any pair of LSR nodes or if the existing connection does not have enough residual bandwidth, then a “possible new link” that does not originate from the destination node or end at the source node will have to be enquired. To construct the final routing topology, the number of lightpaths for which enquiries need to be made is $O(N^2)$ in the worst case, where N is number of LSRs in the network. For a large network, the number of enquiries that need to be made will become prohibitively large, and the large number of lightpath computations caused by the enquiry process will significantly increase the processing overheads in the optical layer and render the scheme impractical for real-time provisioning.

Another important issue that needs to be addressed in finding an optimal route in this approach is how one can avoid finding a false route, i.e. one that cannot be established in the network. This issue arises when the computations for one enquired lightpath is independent from the computation of any other enquired lightpaths. In Figure 7.1, if VL_{24} and VL_{43} are computed independently from each other, the optical layer may find resource for VL_{24} and VL_{43} separately; however, the optical layer may not be able to guarantee that VL_{24} and VL_{43} can be created at the same time. For example, if VL_{24} and VL_{43} compete for the same resource in the optical layer, we may have a situation whereby either VL_{24} or VL_{43} may be created but not both. In this case, the minimum cost route (VL_{12} , VL_{24} , VL_{43}) cannot be set up in the network. It is not difficult to understand why the *false route issue* arises but the problem cannot be solved easily. This is because we cannot possibly ask the optical layer to compute the enquired lightpaths in such a way to ensure that all the new virtual links required in the final routing topology can be created simultaneously. Doing so is tantamount to asking the optical layer to solve the well-known *static lightpath establishment* (SLE) problem, which is known to be computationally intensive and not suitable for real-time provisioning. An alternative solution to this problem is for the optical layer to provide the IP layer information pertaining to the conflicting resources for all the enquired virtual links; however, this may complicate the coordination mechanism of the enquiry process, and may also lead to heavy computational overhead in the next step when IP layer route computation is to be carried out.

It is evident from the above that there is no easy way to find an optimal route in the enquiry-based provisioning scheme. In view of this, we propose two novel routing algorithms for finding LSP routes with near-optimal resource usage. The proposed

algorithms are efficient but not complicated. Our simulations show that these algorithms work well and their performances closely approach to the performance of a peer model based integrated routing algorithm. In both of our proposed algorithms, at most one new virtual link is used in the resultant route so that the false route problem is avoided. In the first proposed algorithm, at most two lightpaths need to be enquired. The second proposed algorithm needs to enquire for at most $2 \times N$ lightpaths, where N is the number of LSRs in the network. We will show later that if the optical layer chooses certain specific routing algorithms, then for the second proposed algorithm, finding all the enquired lightpaths in the optical layer can be easily done with merely two executions of a simple shortest path algorithm.

7.1 Proposed Routing Algorithms

Figure 7.2 Operation of the proposed algorithms

Figure 7.2 illustrates how our proposed algorithms work. In Figure 7.2 a), when a new LSP request reaches the network, the ingress LSR checks its current routing topology and deletes all the existing virtual links that do not have enough residual bandwidth for the new request. Note that if there are more than one virtual links between a node-pair, the one that has enough residual bandwidth with the lowest cost will be selected. This forms a new IP layer routing topology, G_{ip} . Based on this topology, a directed *shortest path tree* (SPT) T_s with source node s as the root and a directed *reversed shortest path tree* (RSPT) RT_d with destination node d as the root can be found. These are shown as solid lines in Figures 7.2 b). In RSPT, root node d can be reached from any other tree node x through a unique path on the tree; the unique path is a directed shortest path in G_{ip} from x to d . Similarly, in SPT, root node s can reach any other tree node x through a unique path on the tree, and the unique path is also a directed shortest path from s to x . For each tree, the cost or distance between a tree node and the root node is recorded. For a tree node k of T_s , its cost to the root node s on tree T_s is denoted as TC_s^k ; similarly, for a tree node k of RT_d , its cost to the root node d on the tree RT_d is denoted as TC_d^k . Based on these two directed trees, for each node x , new directed virtual links are added, as shown in dotted lines, in Figure 7.2 c). Essentially for SPT, we try to add directed virtual links from every tree node to the destination node d , and for RSPT, we try to add directed virtual links from source node s to every tree node. Once we finish adding these directed virtual links to the trees, we store the costs of newly added virtual links in a two-dimension array D . For a newly added virtual link which starts at node m and ends at node n , we use $D_{m,n}$ to denote its cost.

With the final topologies G_{ip-1} and G_{ip-2} formed, we can start to search for the most efficient LSP as follows: Between node pair s and d , there exist only three types of eligible routes on G_{ip-1} and G_{ip-2} : Type 1 is the unique route on the tree that connects nodes s and d , such as $(s \rightarrow 1 \rightarrow 2 \rightarrow d)$ in Figure 7.2 a). Type 2 includes those routes that exist in graph G_{ip-1} in Figure 7.2 c), of which each eligible route reaches the destination node d always via a newly added virtual link (i.e., dotted line). Type 3 consists of those routes that exist in graph G_{ip-2} in Figure 7.2 c), of which each eligible route leaves source node s always via a newly added virtual link. Given the tree node costs TC_s^k and TC_d^k , and an array D that stores the costs of newly added virtual links, we may select the most efficient route from these three types of routes to establish the LSP. The cost of Type 1 route is TC_d^s or TC_s^d , which are the same. Note that if the source node cannot reach the destination node on the original graph G_{ip} , i.e., the source and destination nodes are not on any common tree, then its cost is set to infinite. The costs of Type 2 routes can be described by the set $\{TC_s^k + D_{k,d}, k \in T_s\}$ (see topology G_{ip-1}). Similarly, the costs of Type 3 route can be described as $\{TC_d^k + D_{s,k}, k \in RT_d\}$ (see topology G_{ip-2}). We can select the route with the minimum cost among all the three types of routes to establish the LSP. Though this scheme may not have resulted in the best route being selected, which requires an exhaustive search that is computationally inefficient, it would generally select a good route that would at most results in one virtual link in the optical layer to be used.

Note that our route searching process will only choose a route that does not require any new virtual link to be set up or one new virtual link to be set up. For the

latter case, the new virtual link corresponds to one of the dotted lines shown in Figure 7.2.

In the previous step when G_{ip-1} and G_{ip-2} are constructed, we assume that the cost matrix D is a given parameter. The next step is to determine the value of each unit in the matrix. The different approaches used to obtain the cost matrix D differentiate the two algorithms to be described. We refer to the first algorithm as *integrated routing with estimated link cost* (IRELC) and the second as *integrated routing with tree enquiry* (IRTE).

7.1.1 Integrated Routing with Estimated Link Cost (IRELC)

In this algorithm, each LSR maintains an estimated link cost matrix D . This algorithm is implemented as follows: Whenever a new wavelength-LSP (λ -LSP) between LSR_m and LSR_n is established, the cost of that λ -LSP is disseminated in the IP layer. Each LSR then updates the cost of $D_{m,n}$ in the cost matrix with the new cost. Whenever an existing λ -LSP between LSR_m and LSR_n is released, the cost of that λ -LSP is disseminated in the IP layer. Each LSR then updates the cost of $D_{m,n}$ with the cost of the released λ -LSP. Using this information, an LSR estimates the cost of a future λ -LSP between LSR_m and LSR_n to be $D_{m,n}$, which is the cost of the latest created or released λ -LSP between LSR_m and LSR_n . In other words, historical data is used to estimate the cost of a future λ -LSP.

Note that in the IRELC algorithm, no enquiry process is required at the stage when G_{ip-1} and G_{ip-2} graphs are formed, as we only use the estimated cost matrix D maintained by each LSR to form those graphs. The simplicity of operation is gained at

the expense of LSP establishment which is unable to be guaranteed once its route is computed. If the computed route only contains virtual links that already exist, then the LSP may be established without any problems. However, the algorithm may find a route traversing a new virtual link that has yet to be created; as such, we still need to know if this new virtual link can actually be created by the optical layer. For this reason, an enquiry is sent to the optical layer for the new virtual link. If the enquiry yields a positive response, the LSP is established. If the enquiry results in a negative response, the request is rejected and such a rejection will degrade the efficiency of LSP provisioning.

The aforementioned problem arises because the LSP route is selected based on the estimated distance matrix, which does not precisely reflect the true cost of a new virtual link, and there is no guarantee that a specific new virtual link can actually be created by the optical layer. To improve performance, the following step is added to the IRELC algorithm. When a request reaches the network, the ingress LSR always first tries to establish a new lightpath between the source and destination LSRs, i.e., by using only the resources in the optical layer. If this is successful, the request is served with the new lightpath connecting the source and destination LSRs, and the algorithm terminates. Otherwise, the algorithm continues with the route searching process shown in Figure 7.2 to search for resources in both IP and optical layers which, when combined, would make provisioning possible. With this refinement, the service provisioning process based on the IRELC algorithm may be seen as an improved version of the optical-layer-first provisioning scheme.

7.1.2 Integrated Routing with Tree Enquiry (IRTE)

For the IRTE algorithm, instead of estimating the costs of virtual links, the IP layer sends enquiries to the optical layer to find out the actual costs of the newly added virtual links. For both G_{ip-1} and G_{ip-2} , at most $N-1$ new lightpaths need to be enquired, where N is the total number of LSR nodes in the network. Thus, for the two graphs, at most a total of $2*(N-1)$ lightpaths need to be enquired. The resulting route of the proposed routing algorithm may use at most one new virtual link; this implies that each enquired lightpath may be computed independently from all the other enquired lightpaths since there is no need to create more than one new lightpath at the same time. This property greatly simplifies the process of computing for all the enquired lightpaths in the optical layer.

We notice in G_{ip-1} of Figure 7.2 c) that, in order to fulfill the enquiry requirement the optical layer needs to find the costs of a set of lightpaths with different source OXCs and the same destination OXC_d , and in G_{ip-2} that the optical layer needs to find the costs of a set of lightpaths with the same source OXC_s and different destination OXCs. In other words, if we can find a reverse shortest path tree ORT_d with OXC_d as the root in the optical layer, then the cost of all the enquired lightpaths in G_{ip-1} can be found. Similarly, if we can find a shortest path tree OT_s with OXC_s as the root in the optical layer, then the costs of all the enquired lightpaths in G_{ip-2} can be found. The complexity of finding ORT_d or OT_s is determined by the routing algorithm chosen by the optical layer. If the optical layer chooses a fully adaptive routing algorithm [39], in which each OXC in the optical layer are assumed to maintain up-to-date network state and can compute a shortest path tree with a simple shortest path algorithm (e.g., Dijkstra algorithm), finding ORT_d and OT_s become an easy task. OT_s can be found

with one execution of Dijkstra's algorithm and ORT_d can be found with one execution of a modified version of the standard Dijkstra's algorithm, which will be described shortly. If the optical layer chooses a fix-alternate routing scheme [42], in which the routes for a lightpath demand are predefined and chosen alternately, then finding ORT_d and OT_s will be much more complicated than the fully-adaptive one.

The modification required in the standard Dijkstra algorithm for finding ORT_d is described in what follows. For the standard Dijkstra's algorithm, the initial state consists of the root permanently labeled and assigned a weight of 0, and all the other nodes in G are not labeled and assigned a weight of infinity. The followings are carried out in each iteration of the algorithm:

For a newly labeled node p :

Step 1: If $Weight_p + cost_{pj} < Weight_j$, then $Weight_j \leftarrow Weight_p + cost_{pj}$. $\forall$ (link $p \rightarrow j \in \text{Graph}$). The preceding node of j is set as p , and node j is temporarily labeled.

Step 2: For all the temporarily labeled nodes, find the one with minimum weight, and then permanently label that node.

The iteration will go on until all the nodes are permanently labeled or there is no temporarily labeled node left.

We only need to modify step 1 of the standard Dijkstra algorithm in order to find a RSPT:

New step1: If $Weight_p + cost_{jp} < Weight_j$, then $Weight_j \leftarrow Weight_p + cost_{jp}$. $\forall$ (link $j \rightarrow p \in \text{Graph}$). The succeeding node of j is set as p , node j is temporarily labeled.

For both algorithms, finding a SPT in the IP layer can be easily done by one execution of Dijkstra's algorithm. The RSPT in the IP layer can also be found with one execution of the modified Dijkstra algorithm. At most two executions of Dijkstra's algorithm are needed in the IP layer. Compared with the IRELC algorithm, the complexity of the IRTE algorithm is mainly decided by the complexity of finding the shortest path tree in the optical layer. If finding a SPT in the optical layer is too complicated, then the IRELC algorithm is a better choice because of its simplicity (i.e., it only needs to enquire at most two lightpaths - the first one is for the lightpath connecting LSR_s and LSR_d at the beginning of the algorithm and the second one is needed in case the resultant route yielded by the searching process depicted in Figure 7.2 needs to use a new virtual link). Otherwise, IRTE can be considered since our simulation results show that the IRTE strategy can yield a performance which is almost as good as the one based on the integrated routing approach of the peer model.

7.2 Simulations

We evaluate the performance of the proposed algorithms by means of simulation. Simulations are carried out using the ARPANet topology described in Chapter 4. ARPANet has 20 nodes and 64 directional links. The number of wavelengths per link is set to 8 and we assume that the network has no wavelength conversion capability. We have also run simulations for other network topologies (e.g., for Europe network with 33 nodes and 136 directional links [58]) and the results are similar.

For the simulation, LSP requests are generated in accordance with a Poisson process with mean arrival rate λ . The holding time of an LSP request is assumed to be negative exponentially distributed with unity mean, i.e., $1/\mu = 1$. A request is rejected

if the network resource is inadequate to give it the required bandwidth. The source and destination nodes for an LSP request are randomly assigned. As in [50][51], we evaluate the system performance using the *request blocking ratio* (RBR), which is defined as

$$\text{RBR} = \frac{\sum_i \text{BRB}_i}{\sum_j \text{RB}_j} \quad (7-1)$$

where BRB_i is the bandwidth of the i th blocked request and RB_j is the bandwidth of the j th request.

We compare our proposed algorithms with those presented in [50][51]. Since [50] shows that the optical-first algorithm significantly outperforms the IP-first algorithm, we only use the optical-first algorithm in our comparisons here. For the peer model, integrated routing is carried out as in [48]. In the simulation of IRELC and IRTE, the optical layer chooses full-adaptive routing and wavelength assignment [58]. We run each of our simulation experiments for a duration such that at least 2×10^6 LSP requests arrive into the system.

7.2.1 Effect of Bandwidth Granularities of LSP Requests

We consider a situation in which all the incoming LSP requests have the same bandwidth requirement. The capacity of a lightpath is set to unity. The results obtained for different bandwidth granularities of LSP requests (i.e., different LSP request bandwidths) are shown in Figure 7.3, with RBR plotted against the load of LSP requests.

(a) LSP request bandwidth 0.1

(b) LSP request bandwidth 0.2

(c) LSP request bandwidth 0.3

(d) LSP request bandwidth 0.4

(e) LSP request bandwidth 0.5

Figure 7.3 Effect of bandwidth granularity of LSP requests

As expected, for all the bandwidth granularities considered, integrated routing of the peer model achieves the best RBR performance, optical-first algorithm (overlay model) is the worst while our proposed IRTE algorithm works almost as well as integrated routing. The proposed IRELC algorithm has poorer performance compared to IRTE but has better performance compared to optical-first.

When the LSP request bandwidth is low (i.e., low bandwidth granularity), the performance differences among the four algorithms are small. However, as the bandwidth granularity becomes larger, e.g., changes from 0.3 to 0.5, there are significant performance differences among these algorithms. When the LSP bandwidth granularity is larger than 0.5 (not shown in Figure 7.3), the four algorithms

yield almost the same results because the virtual link can no longer groom any of the low-order LSPs. The result suggests that if the bandwidth granularity is small, the operator may choose optical-first because of its simplicity; however, when the bandwidth granularity is larger, IRTE or IRELC may be a better choice overall because peer model is complex and costly.

The overall traffic load of the network is defined as $B^* \lambda / \mu$ Erlang, where B is the bandwidth requirement of an LSP request. We find that for the same traffic load, when B is different, network performance is different. These results are shown in Figure 7.4 a), which plots RBR against the LSP request traffic for different values of B . We only present the result of IRTE since the results of the other algorithms shows similar trends. We notice that for the same traffic load, as bandwidth granularity B increases, the performance of the system deteriorates. This phenomenon can be explained as follows: As the bandwidth requirement of an LSP request increases, there is less chance that it can use the existing virtual links to route the request. The request then tends to be routed by setting up a new lightpath which uses up more resources from the optical layer. As more resources are used up for the current LSPs, the blocking probability for new requests increases. This explanation is supported by the statistical data collected in the simulation, which is shown in Figure 7.4 b). The figure shows the average number of lightpaths that exist in the network versus the LSP request traffic for different values of B . As can be seen, for the same traffic load, the average number of lightpaths is larger for higher values of B .

(a) RBR performance

(b) Average number of lightpaths

Figure 7.4 Performance results for different granularities

7.2.2 Performance Comparison for Dynamically Varying Bandwidth Requirement

In real-life, the bandwidth requirements of incoming LSP requests will vary from request to request instead of being homogeneous. To test our proposed algorithms under dynamically varying bandwidth requirements, we assume that the bandwidth requirements of LSP requests are random variables uniformly distributed over certain range. In the simulation, this range is set to be either (0.1, 0.3) or (0.1, 0.5) so that the average bandwidth granularity is 0.2 or 0.3, respectively. Figure 7.5 shows the performance of the algorithms under these conditions for both cases. As expected, integrated routing for the peer model still has the best performance and optical-first routing the worst. The proposed IRTE algorithm works almost as well as integrated routing for the peer model. The IRELC algorithm shows poorer performance but still performs better than the optical-first algorithm. As observed earlier, the performance differences between the various schemes widen with increasing bandwidth granularities.

Figure 7.5 Performance comparison for different dynamic bandwidth granularities

Figure 7.6 shows a similar performance comparison. We consider the performance when (a) bandwidth granularity is constant at 0.3, (b) varies uniformly in the range (0.2, 0.4), and (c) varies uniformly in the range (0.1, 0.5). Note that the performance becomes poorer with increased randomness even though the mean remains the same at 0.3. This is because an LSP request with a larger bandwidth requirement is more likely to be blocked than one with a lower bandwidth requirement. When the range of the LSP bandwidth request is wide and the distribution is uniform across the entire range, the proportion of high bandwidth LSP requests increases and more LSPs are blocked.

Figure 7.6 Performance comparison for different static and dynamic bandwidth granularities

7.3 Conclusions and discussions

In this Chapter, we propose two integrated routing algorithms IRELC (*Integrated Routing with Estimated Link Cost*) and IRTE (*Integrated Routing with Tree Enquiry*) to be used in conjunction with the enquiry process. Compared with integrated routing for the peer model, IRELC and IRTE are much simpler in terms of control plane design and operation but can achieve almost the same performance. We study the effect of bandwidth granularity of LSP requests on network performance for the various schemes and observe that performance variation is greater for bandwidth granularities between 0.3 and 0.5. We also observe that fluctuation in bandwidth requirements of LSP requests have a degrading effect on system performance.

In our study, we define the cost of a virtual link to be the number of wavelength channels traversed in the optical layer; hence, the minimum cost LSP found is the one which utilizes the least number of hops traversed in the optical layer. If transmission delay is an important concern to a network operator, the cost of a virtual link can be set to reflect the delay in the optical layer. In such a scenario, finding the minimum cost LSP translates into finding an LSP with the least transmission delay. We expect this way of equating cost will increase the blocking rate since the least cost LSP is no longer one that uses the least number of wavelength channels.

Another factor that may be of interest is the propagation delay incurred by signaling messages. In our study, we assume that each LSR and OXC maintains update-to-date network status information. If the network status information is out-of-date due to signaling message delay, obviously, false routing may rise and lead to non-optimal routing.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

Wavelength routed optical networks appear to be an attractive technology option for implementing wide-area and metropolitan-area transport networks. Finding a suitable *routing and wavelength assignment* (RWA) algorithm is a very important problem that needs to be solved in the design of a wavelength routed optical network. RWA related design problems could be classified into four categories in accordance with the types of traffic pattern and the stage of the design and planning process; they are, namely, *static lightpath establishment* (SLE) problem, *dynamic lightpath establishment* (DLE) problem, *virtual topology design* (VTD) problem, and *dynamic LSP provisioning* (DLP) problem.

Though SLE, DLE, VTD, and DLP routing and wavelength assignment problems have been widely studied individually, there is no cohesive study of all the aforementioned problem domains in a collective manner. To fill that void, in Chapter 3, we provide an extensive review of different representative solution techniques for the above-mentioned problems. Also, in studying the SLE and DLP problems, we devise better solution techniques that outperform the best-known solutions in one or more ways. Those solutions are described in Chapter 4-7.

For the SLE problems, most solutions proposed in past studies are considered computationally expensive, especially when large networks are concerned. For this reason, we propose a *tabu search* (TS) algorithm in Chapter 4 to solve the well-known *maximum lightpath establishment* (MLE) problem. The proposed TS algorithm has the salient feature that routing and wavelength assignment are considered jointly to achieve greater optimality while keeping computing requirement low. Since we do not use more sophisticated and complicated TS techniques such as long-term memory utilization, path re-linking, strategic oscillation, the proposed TS algorithm is quite easy to implement. We compare the proposed TS algorithm with ILP formulation, which sets the upper bound for optimality, and the results for both small and large networks show that the proposed TS algorithm works almost as well as the ILP solution and is much more computationally efficient. We also propose a heuristic solution based on the proposed MLE TS algorithm for solving the *minimum wavelength usage* (MWU) problem -- another important problem in SLE field. Simulation results show that our heuristic solution works very well in term of optimality and computational cost. Since our TS algorithm can solve the SLE problem within reasonable time (within half an hour for a 50 node network in our simulation), it is practical for network operators to use our TS algorithm for network design in real-life.

Most network operators are mainly interested in maximizing profit using the existing network facilities. For this reason, we provide a new problem formulation in Chapter 5 for the SLE problem aiming at maximizing the revenue of an optical network. To model the presence of two types of requests in real-life, the lightpath requests are assumed to have two different priority levels. We prove that, with some

minor modifications, the MLE TS algorithm is suitable for solving the *maximum revenue* (MR) problem. An ILP solution and a simple *sequential RWA* (SR) algorithm are also proposed. The simulation results show that the SR algorithm is more computationally efficient. In term of optimality of solution found, the TS algorithm is only slightly inferior to the ILP solution and much better than the SR algorithm.

For the DLP problem, we propose in Chapter 6 an enquiry process that can be used by the IP layer to obtain from the optical layer information on the availability of resource for setting up certain lighpaths. Different enquiry schemes and the corresponding message formats are described. The proposed enquiry process has the merit of not violating the layer independency feature of the overlay model but able to obtain useful availability information on optical layer resource to allow for better resource allocation and utilization.

In Chapter 7, two integrated routing algorithms IRELC (*integrated routing with estimated link cost*) and IRTE (*integrated routing with tree enquiry*) are proposed to be used in conjunction with the enquiry process for achieving near optimal capacity usage while avoiding excessive signaling and control overhead arising from the enquiry process. The proposed algorithms perform significantly better than other routing algorithms for the overlay model and are nearly as good as the integrated routing algorithm for the peer model.

8.2 Suggestions for Further Research

In this dissertation, we assume that there is no wavelength converter in the network; thus, the wavelength continuity constraint has to be strictly followed. Our

assumption is based on two concerns: 1) The presence of O-E-O wavelength converters will impair the optical layer transparency pertaining to higher layer data format and encoding; 2) All-optical wavelength converters will not be commercially viable at least for the next few years. However, as technology advances, optical networks will support some degree of wavelength conversions eventually. Clearly, TS search algorithm for such a network will be a logical extension of the present work.

Optical network is prone to failures. Due to the huge amount of bandwidth that is possible with the WDM technique, a single fiber cut will affect multi-terabit-per-second of traffic and a node failure will affect even more. Naturally network survivability is a major concern of network service providers. Resiliency is normally achieved by means of protection or restoration. For a protection scheme, normally two lightpaths are established for each request: One is the working lightpath that carries the traffic before the failure occurs; the other is a backup lightpath that will take over when the working lightpath is down due to failure. To make the protection scheme work, it is essential that the working and backup lightpaths are not affected by a failure simultaneously. This scenario is different from the RWA problem discussed in this dissertation, in which each request only needs one lightpath. A lot of work is required to modify our TS algorithm and enquiry based algorithm to cater for protection, which makes this a suitable topic for further research.

List of Author's Publications

1. Y. Wang and T. H. Cheng, "Dynamic routing and wavelength assignment scheme for protection against node failure," Proc. of Globecom 2003. IEEE, Vol. 5, pp. 2585 – 258, Dec. 2003.
2. Y. Wang, T. H. Cheng and S. K. Bose, "An enquiry-based dynamic service provisioning approach for overlay IP-over-optical networks," Photonics Network Communications, Vol. 10, No. 3, pp347-360, Nov. 2005.
3. Y. Wang, T. H. Cheng and M. H. Lim, "A Tabu search algorithm for static routing and wavelength assignment problem," IEEE Communications Letters, Vol. 9, No.9, pp841-843, Sept. 2005.
4. Y. Wang, T. H. Cheng and Maode Ma, "Priority and Maximum Revenue based Routing and Wavelength Assignment for All-optical WDM Networks," IEEE International Conference on Research, Innovation and Vision for the Future 2007, Vol. 5-9, pp135 – 139, Mar.2007

References

- [1] P. E. Green, "Optical Networking Update," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 5, pp. 764-779, Jun. 1996.
- [2] P. E. Green, "Progress in Optical Networking," IEEE Communications Magazine, Vol.39, No. 1, pp. 54-61, Jan. 2001.
- [3] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, Moragn Kaufmann Publishers Inc. 1997.
- [4] B. Rajagopalan, J. Luciani, and D. Awduche, "IP over Optical Networks: A Framework," IETF Draft draft-ietf-ipo-framework-04.txt, Aug. 2003.
- [5] H. Zang, J. P. Jue, and B. Mukherjee, "Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks," Optical Networks Magazine, Vol. 1, No.1, pp. 47-60, Jan. 2000.
- [6] D. Colle, S. De Maesschalck, C. Develder, P. Van Heuven, A. Groebbens, *et al*, "Data-centric Optical Networks and Their Survivability," IEEE Journal on Selected Areas in Communications, Vol. 20, No. 1, pp. 6 – 20, Jan. 2002.
- [7] N. Ghani, S. Dixit and T. S. Wang, "On IP-over-WDM Integration," IEEE Communications Magazine, Vol. 38, No. 3, pp. 72 – 84, Mar. 2000.
- [8] R. Ramaswami and K. Sivarajan, "Design of Logical Topologies for Wavelength Routed Optical Networks," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 5, pp.840-851, Jun. 1996.
- [9] R. Krishnaswamy and K. Sivarajan, "Design of Logical Topologies: a Linear Formulation for Wavelength Routed Optical Networks with no Wavelength Changers," IEEE/ACM Transactions on Networking, Vol. 9, No.2, pp. 186-198, Apr. 2001.
- [10] T. D. Nadeau, C. Srinivasan and A. Farrel, "Multi-protocol Label Switching (MPLS) Management Overview," IETF Draft draft-ietf-mpls-mgmt-overview-09.txt, Sep. 2003.
- [11] L. Berger, "Generalized MPLS - Signaling Functional Description," IETF RFC 3471, Jan. 2003.

- [12] G. Jeong and E. Ayanoglu, "Comparison of Wavelength-Interchanging and Wavelength-Selective Cross-connects in Multi-wavelength All-Optical Networks," Proc. of INFOCOM 1996, Vol. 1, pp. 156-163, Mar. 1996.
- [13] R. A. Barry and P. A. Humblet, "Models of Blocking Probability in All-Optical Networks with and without Wavelength Changers," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 5, pp. 858–867, Jun. 1996.
- [14] M. Kovačević and A. S. Acampora, "Benefits of Wavelength Translation in All-Optical Clear-Channel Networks," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 5, pp. 868–880, Jun. 1996.
- [15] S. John, A. L. Chiu and R. Tkach, "Issues For Routing In The Optical Layer," IEEE Communications Magazine, Vol. 39, No.2, pp. 81-87, Feb. 2001.
- [16] L. Andersson, P. Doolan, N. Feldman, A. Fredette and B. Thomas, "LDP Specification," IETF RFC 3036, Jan. 2001.
- [17] R. Braden, L. Zhang, S. Berson S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP)- version 1 Functional Specification," IETF RFC 2205, Sep. 1997.
- [18] K. Kompella and Y. Rekhter, "LSP Hierarchy with Generalized MPLS TE," IETF Draft draft-ietf-mpls-lsp-hierarchy-08.txt, Mar. 2002.
- [19] G. Li, J. Yates, D. Wang, and C. Kalmanek, "Control Plane Design for Reliable Optical Networks," IEEE Communications Magazine, Vol. 40, No. 2, pp.90 – 96, Feb. 2002.
- [20] J. Moy, "OSPF Version 2," IETF RFC 2328, Apr. 1998.
- [21] D. Oran, "OSI IS-IS Intra-Domain Routing Protocol," IETF RFC 1142, Feb. 1990
- [22] J. Lang, "Link Management Protocol (LMP)," IETF Draft draft-ietf-ccamp-lmp-10.txt, Oct. 2003.
- [23] D. Awduche, D. Gan, T. Li V. Srinivasan and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF RFC 3209, Dec. 2001.
- [24] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu *et al.* "Constraint-based LSP setup Using LDP," IETF RFC 3212, Jan. 2002.
- [25] D. Banerjee and B. Mukherjee, "A Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks," IEEE Journal on Selected Areas in Communications, Vol. 14, no. 5, pp. 903-908, Jun. 1996.
- [26] K. Lee and K. Choe, "Routing and Wavelength Assignment in WDM All-Optical Networks," IEEE Electronics Letters, Vol. 36, No.11, pp. 964-966, May 2000.

- [27] N. Banerjee and S. Sharan, "A Evolutionary Algorithm for Solving the Single Objective Static Routing and Wavelength Assignment Problem in WDM Networks," Proc. of IEEE ISIP 2004, pp. 13-18, Jan. 2004.
- [28] R. Ramaswami and K. Sivarajan, "Routing and Wavelength Assignment in All-Optical Networks," IEEE/ACM Transactions on Networking, Vol.3, No.5, pp. 489-500, Oct. 1995.
- [29] R. Krishnaswamy and K. Sivarajan, "Algorithms for Routing and Wavelength Assignment Based on Solution of LP-Relaxation," IEEE Communications Letters, Vol. 5, no.10, pp.435-437, Oct. 2001.
- [30] P. Aiyarak, A. S. Saket and M. C. Sinclair, "Genetic Programming Approaches for Minimum Cost Topology Optimization of Optical Telecommunication Networks," Proc. of IEEE GALEZIA 97, pp. 415-420, Sep. 1997.
- [31] H. W. Tzeng, J. L. Chen and N. K. Chen, "Traffic Grooming in WDM Networks Using Genetic Algorithm," Proc. of IEEE SMC '99, Vol.1, pp. 1003-1006, Oct. 1999.
- [32] Glover F., "Future paths for Integer Programming and Links to Artificial Intelligence," Computers and Operations Research, Vol. 13, No.5, pp. 533-549, May 1986.
- [33] A. Hertz and D. D. Werra, "Using Tabu Search Techniques for Graph Coloring," Computing, Vol. 39, No. 4, pp.345--351, Dec. 1987.
- [34] A M. Widmer, "The Job-shop Scheduling with Tooling Constraints: A Tabu Search Approach", Journal of Operational Research Society, Vol. 42, No.1, pp. 75-82, Jan. 1991.
- [35] S. Yan, M. Ali and J. Deogun, "Route Optimization of Multicast Sessions in Sparse Light-splitting optical networks," Proc. of IEEE GLOBECOM 2001, Vol. 4, pp. 2134-2138, Nov. 2001.
- [36] G. D. Morley, W. D. Grover, "Tabu Search Optimization of Optical Ring Transport Networks," Proc. of IEEE GLOBECOM 2001, Vol. 4, pp. 2160-2164, Nov. 2001.
- [37] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro and R. Douville, "Routing and Wavelength Assignment of Scheduled Lightpath Demands", IEEE Journal on Selected Areas in Communications, Vol. 21, no. 8, pp. 1231-1240, Oct. 2003.
- [38] A. Mokhtar and M. Azizoglu, "Adaptive Wavelength Routing in All-Optical Networks," IEEE/ACM Transactions on Networking, Vol. 6, No.2, pp. 197-206, Apr. 1998.

- [39] C. Chen and S. Banerjee, "A New Model for Optimal Routing and Wavelength Assignment in Wavelength Division Multiplexed Optical Networks," Proc. of IEEE INFOCOM 1996, Vol. 1, pp. 164-171, Mar. 1996.
- [40] C. A. Farago and T. Zhang, "Lightpath (Wavelength) Routing in Large WDM Networks," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 5, pp. 909-913, June 1996.
- [41] H. Harai, M. Murata, and H. Miyahara, "Performance of Alternate Routing Methods in All-Optical Switching Networks," Proc. of IEEE INFOCOM 1997, pp. 516-524, Apr. 1997.
- [42] S. Ramamurthy and B. Mukherjee, "Fixed-Alternate Routing and Wavelength Conversion in Wavelength-Routed Optical Networks," Proc. of IEEE GLOBECOM 1998, pp. 2295-2302, Nov. 1998.
- [43] L. Li and A. K. Somani, "Dynamic Wavelength Routing Using Congestion and Neighborhood Information," IEEE/ACM Transactions on Networking, Vol.7, No.5 pp. 779-786, Oct. 1999.
- [44] J. P. Jue and G. X. Xiao, "An Adaptive Routing Algorithm with a Distributed Control Scheme for Wavelength-Routed Optical Networks," Proc. of IEEE CCN 2000, pp. 192-197, Oct. 2000.
- [45] A. Birman and A. Kershenbaum "Routing and Wavelength Assignment Methods in Single-Hop All-Optical Networks with Blocking," Proc. of IEEE INFOCOM 1995, Vol. 2, pp. 431-438, Apr. 1995.
- [46] D. Banerjee and B. Mukherjee, "Wavelength-routed Optical Networks: Linear Formulation, Resource Budgeting Tradeoffs, and a Reconfiguration Study," IEEE/ACM Transactions on Networking, Vol. 8, No.5, pp.598-607, Oct 2000.
- [47] K. Zhu and B. Mukherjee, "Traffic Grooming in an Optical WDM Mesh Network," IEEE Journal on Selected Areas in Communications, Vol. 20, NO.1, pp. 122-133, Jan. 2002.
- [48] M. Kodialam and T.V. Lakshman, "Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks," Proc. of IEEE INFOCOM 2001. Vol. 1, pp. 358-366, Apr. 2001.
- [49] C. Assi, A. Shami, M. A. 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, pp. 36 – 45, Jul.-Aug. 2001.
- [50] W. Zhong, X. Niu and T. H. Cheng, "Comparison of Overlay and Peer Models in IP over Optical Networks," Proc. of COIN 2002, pp. 155-157, Jul. 2002.

- [51] K. Zhu and B. Mukherjee, "On-line Approach for Provisioning Connections of Different Bandwidth Granularities In WDM Mesh Network," Proc. of IEEE OFC 2002, pp. 549-551, March 2002.
- [52] A. Khalil, A. Hadjiantonis, G. Ellinas and M. Ali, "A Novel IP-Over-Optical Network Interconnection Model for the Next-generation Optical Internet", Proc. of IEEE GLOBECOM 2003, Vol. 7, pp. 3984-3989, Dec. 2003.
- [53] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1998.
- [54] S. M. Sait and H. Youssef, *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*, IEEE Computer Society, 1999.
- [55] J. Y. Yen, "Finding the K shortest Loopless Paths in a Network," Management Science, Vol. 17, no. 11, pp. 712-716, Jul. 1971.
- [56] S. Baroni and P. Bayvel. "Wavelength Requirements in Arbitrarily Connected Wavelength- Routed Optical Networks", IEEE Journal of Lightwave Technology, Vol. 15, No. 2, pp. 242-251, Feb. 1997.
- [57] S. Ohta and A. Greca, "Comparison of Routing and Wavelength Assignment Algorithms for Optical Networks," IEEE Workshop on High Performance Switching and Routing, pp.146-149, May 2001.
- [58] W. Grover, J. Doucette, M. Clouqueur, D. Leung and D. Stamatelakis, "New Options and Insights for Survivable Transport Networks," IEEE Communications Magazine, Vol. 40, No. 1, pp. 34-41, Jan. 2002.
- [59] Mohamed Saad and Zhi-Quan Luo,"On the Routing and Wavelength Assignment in Multifiber WDM Networks," IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Vol. 22, No. 9, pp. 1708-1717, Nov. 2004
- [60] Steven S. W. Lee, Maria C. Yuang, Po-Lung Tien, and Shih-Hsun Lin," A Lagrangean Relaxation-Based Approach for Routing and Wavelength Assignment in Multigranularity Optical WDM Networks," IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Vol. 22, No. 9, pp. 1741-1751, Nov. 2004
- [61] Yiming Zhang, Oliver Yang and Haomei Liu,"A Lagrangean Relaxation and Subgradient Framework for the Routing and Wavelength Assignment Problem in WDM Networks," IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Vol. 22, No. 9, pp. 1752-1765, Nov. 2004
- [62] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

Acronyms

AS	autonomous system
ATM	asynchronous transfer mode
AVS	assigned variable set
CAVS	number of variables in AVS
CPS	candidate path set
DLE	dynamic lightpath establishment
DLP	dynamic labeled switched path provisionin
ERM	enquiry response message
FA	forward adjacency
FAA	full adaptive algorithm
FEC	forwarding equivalent class
FSC_LSR	fiber-switching capable label switching router
FPLC	fixed-path least-congestion
GA	genetic algorithms
GMPLS	generalized multi-protocol label switching
HPS	high priority set
IETF	Internet engineering task force
ILP	integer linear programming
IRA	Integrated Routing Algorithms
IRELC	integrated routing with estimated link cost
IRTE	integrated routing with tree enquiry
LCC	lightpath conflict constrain
LDP	label distribution protocol
LILP	link-based integer linear programming formulation
LMP	link management protocol
LPS	low priority set
LSC-LSR	lambda-switching capable label switching router
LSP	labeled switched path
LSR	label switching router
MAN	wavelength division multiplexing
MPLS	multi-protocol label switching
MLE	maximum lighthpath establishment
MR	maximum revenue
MWU	minimum wavelength usage

NNI	network-to-network interfaces
OXC	optical cross-connects
PILP	path-based linear programming formulation
PSC-LSR	packet-switching capable label switching router
RBR	request blocking ratio
RMR	relaxed maximum revenue
RSPT	reversed shortest path tree
RSVP	resource reservation protocol
RWA	routing and wavelength assignment
SDH	synchronous digital hierarchy
SLE	static lightpath establishment
SPT	shortest path tree
SR	sequential routing and wavelength assignment
TS	tabu search
TSC-LSR	TDM-switching capable label switching router
UNI	user-to-network interfaces
VLEM	virtual link enquiry message
VTD	virtual topology design
WCC	wavelength continuity constraint
WDM	wavelength division multiplexing
WIXC	wavelength-interchanging crossconnect
WRON	wavelength routed optical network
WSXC	wavelength-selective crossconnect
WVS	waiting variable set