uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa
L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Shen Yu
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical and Computer Engineering)
GRADE / DEGREE

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

Performance Evaluation of Optimized Resource Allocations in WDM Networks under Dynamic Traffic
TITRE DE LA THÈSE / TITLE OF THESIS

G.V. Bochman
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

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

C. Huang

D. Makrakis

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

Performance Evaluation of Optimized Resource Allocations in WDM Networks under Dynamic Traffic

Shen Yu

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the Degree of

Master of Applied Science, Electrical and Computer Engineering

November 17, 2010

Ottawa-Carleton Institute of Electrical and Computer Engineering

School of Information Technology and Engineering

University Of Ottawa

Ottawa, Ontario, Canada

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-74158-0
Our file Notre référence
ISBN: 978-0-494-74158-0

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Wavelength-routed Wavelength-Division Multiplexing (WDM) optical networks are developed to meet the increasing bandwidth demands of recent Internet applications. In the real world, the traffic demands are always dynamic. The question of how to optimally allocate the network resources in a wavelength-routed WDM optical network to satisfy dynamic demands is called the dynamic routing and wavelength assignment (RWA) problem. In the literature, many schemes are proposed to address this problem.

In this thesis, we study how to improve the performance of certain dynamic RWA schemes by using the distribution of the dynamic demands, which is assumed to be known in advance. With the knowledge of the network topology and distribution of the dynamic traffic, we can use an existing mathematical approach to calculate, for each link, a cost value called Multiplier Guide. This link cost is obtained by considering a static RWA optimization problem having the average traffic demands of the dynamic RWA problem that we want to solve. When this link cost is used instead of the hop count to compute minimal cost paths, the performance of some dynamic RWA schemes can be improved, as shown by our simulation studies.

In the thesis, we review and implement several existing dynamic RWA schemes; we also propose and implement some new dynamic RWA schemes. We apply the Multiplier Guide to all these schemes and run simulations to test whether their performance is improved. As a necessary part of these schemes, we implement many different network routing algorithms including single shortest path routing, link-disjoint paths routing and k-shortest paths algorithm. To simulate the schemes that search an RWA solution exhaustively with or without the wavelength conversion assumption, we implement two auxiliary graphs and two algorithms associated with the graphs, respectively. We also implement the schemes that use link utilization obtained from simulations or

measurements to decide link cost; and we compare them with the schemes that use the Multiplier Guide as link cost.

We find that, when the Multiplier Guide is applied to the dynamic RWA schemes that use pre-constructed paths, their performance is improved. However, when the Multiplier Guide is applied to the schemes that search an RWA solution exhaustively, their performance is not improved.

Acknowledgement

First, I would like to show my deepest appreciation to my supervisors, Dr. Gregor Bochmann and Dr. Jing Wu, for their great academic guidance and financial support during my studies.

Second, I would like to thank our research group member Dr. James Yiming Zhang for developing the theoretical background of my research.

Third, I would like to thank all my friends, the professors and the staff at the University of Ottawa for the help they gave me.

Finally, I would like to thank all my family members for their support.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

WDM	Wavelength-Division Multiplexing
RWA	Routing and Wavelength Assignment
SPR-FF	Shortest Path Routing First-Fit
FAR-FF	Fixed Alternative Routing First-Fit
LLR-FF	Least Loaded Routing First-Fit
HC	Hop Count
MG	Multiplier Guide
BSR	Best among the Shortest Routes
ULULC	Using Link Utilization as Link Cost
M_Dijkstra	Modified Dijkstra
AUR	Adaptive Unconstrained Routing
KSP-FF	K-Shortest Paths Routing First-Fit
Algorithm APTP	Algorithm for Adding a Path to the Tree of Paths
ECSP-FF	Equal Cost Shortest Paths First-Fit
SPAWG	Shortest Path Algorithm in Wavelength Graph
SPR-WC	Shortest Path Routing with Wavelength Conversion
DPR-WC	Disjoint Paths Routing with Wavelength Conversion

Chapter 1

Introduction

1.1 Context of the Thesis

We are in a world of information explosion. The Internet connects us all and handles all the information we generate. Many of the current emerging Internet multimedia applications, such as video conferencing, high-definition video broadcasting and online games, all have enormous bandwidth requirements. The bandwidth requirement in the current networks is very huge and it is still growing. In the Internet backbone, an enormous amount of data is aggregated at each node and needs to be transmitted to its destinations.

Wavelength-routed Wavelength-Division Multiplexing (WDM) optical networks are the ideal candidate for the next generation of networks. With the WDM technology, a large number of wavelength channels can be used concurrently in one single fiber for information transmission. The bandwidth of one fiber is equivalent to that of multiple electrical wires. A WDM network consists of WDM routers interconnected by fibers. A WDM network is the ideal structure for the Internet backbone.

In a wavelength-routed WDM network, point-to-point connections between nodes are used for communications. When a connection between two nodes needs to be established, if no wavelength converter can be used, a path needs to be found first and then a common free wavelength along that path needs to be chosen. This is the well known wavelength-continuity constraint. Wavelength conversion is another important concept in wavelength-routed WDM networks. A node equipped with wavelength converters can convert an incoming signal from one wavelength to another, and send it on the other wavelength. With wavelength converters installed in the network, the wavelength-continuity constraint can be relaxed. In a given network, when information such as network topology, number of wavelengths in each fiber, the possibility of wavelength conversion and the user demands are all known, the problem of how to establish these point-to-point connections is called the Routing and Wavelength Assignment (RWA) problem. It is a well-known problem and studied extensively.

In the real world, the aggregated data at each node to be transmitted to other nodes is not always fixed. As a result, the demands for point-to-point connections arrive at random times, and the connections remain in the network for random durations until they are finally released. How to optimally establish the dynamic connections in a given situation is an important research topic; it is called the dynamic RWA problem.

In the literature, many dynamic RWA schemes are introduced to address this problem. However, these schemes only consider the network topology information. The traffic information, which is the probability distribution of these dynamic demands, is not taken into account in those schemes. Most of these schemes assume that the demands are uniformly distributed in the network. Some of the schemes may assume non-uniform demand distribution, but these schemes still do not make use of this traffic information. So far, schemes that consider traffic information as part of the factors for connection establishment have not been proposed.

1.2 Objective of the Thesis

To meet the increasing bandwidth demands of the Internet applications, wavelength-routed WDM networks are deployed. The problem of how to allocate resources in the network to accommodate the dynamic demands needs to be addressed. The objective of this thesis is to provide some dynamic RWA schemes that can optimally establish the connections for the dynamic demands and achieve very good network performance.

To achieve the objective, we study how the traffic information can be used as a guide to develop better schemes. We assume that the traffic demand distribution is known, and in general non-uniform. In contrast to the existing dynamic RWA schemes, our schemes consider both traffic information and network topology information.

In many RWA schemes, the number of links in a path, called hop count, is used as a measure of the cost of the path; in this case each link has the same cost. We explore in this thesis the possibility of using link costs that take into account the expected congestion on each link. In particular, we use Langrange Multipliers obtained from the RWA optimization performed for static traffic allocation [1]. These Lagrange Multipliers are added as cost to each link. We call this approach to the determination of link costs the Multiplier Guide. In this thesis, we review and implement many different existing dynamic RWA schemes for wavelength-routed WDM networks. Then we apply the Multiplier Guide to these schemes and study how the network performance is improved.

1.3 Organization of the Thesis

The thesis is organized as follows. Chapter 2 gives an introduction about the wavelength-routed WDM network and the related routing and wavelength assignment problems. Chapter 3 describes the theoretical background of our work: the Multiplier Guide. We explain the meaning of the Multiplier Guide and how it can be applied in dynamic RWA. Chapter 4 explains the simulation framework we built for testing different dynamic RWA schemes. Chapter 5 presents the research results of applying the Multiplier Guide to two-stage allocation schemes: Shortest Path Routing First-Fit, Fixed Alternative Routing First-Fit and Least Loaded Routing First-Fit. In this chapter we also study three schemes that set the link cost by using the link utilization obtained from simulations. Chapter 6 presents the results of applying the Multiplier Guide to a one-stage allocation scheme called M_Dijkstra Algorithm. Chapter 7 introduces the method of finding k shortest paths in a graph and then presents the result of applying the Multiplier Guide to the K-Shortest Paths Routing First-Fit scheme. Chapter 8 presents the results of applying the Multiplier Guide to several dynamic RWA schemes that include the consideration of wavelength conversion.

Chapter 2

Connection Establishment in Wavelength-Routed WDM Networks

2.1 Wavelength-Division Multiplexing (WDM)

Wavelength-Division Multiplexing is a promising technology for high-bandwidth data transmission [2]. It is a form of Frequency-Division Multiplexing. Signals carried by light of different frequencies (or different wavelengths) are independent. These signals can be multiplexed and transmitted through a fiber. At the receiving end, each signal can be extracted from the mixed signal. When the WDM technology is applied, in a single fiber, several communication channels can be established simultaneously using different wavelengths without interfering each other. In addition, data transmission in both directions can be carried out simultaneously. This means that the bandwidth is multiplied and high bandwidth transmission over a single fiber is possible.

Depending on how the wavelength space is divided into individual wavelength channels, the WDM technology can be classified into two categories: Coarse WDM and Dense WDM. The Coarse WDM technology (ITU standard G.694.2) uses relatively fewer wavelength channels. Such technology uses around 4 to 8 wavelengths and a single channel has a bandwidth of around 2.5 Gb/s. The data rate of Coarse WDM technology is suitable for communications within metropolitan areas. Compared to the Coarse WDM technology, the Dense WDM technology (ITU standard G.694.1) uses much more wavelength channels. In the Dense WDM technology, the wavelength space can be divided into up to 160 individual wavelength channels. Each individual channel has a bandwidth of 1 to 10 Gb/s and it can be expanded to 40 Gb/s in the future. Dense WDM technology is suitable for the Internet backbone, where the bandwidth of the data flow is immense.

A WDM network is a network of WDM routers interconnected by fibers. A WDM router works as a switch when data passes through. Due to the WDM router's structure, a signal in one wavelength channel coming into a WDM router can be sent out through one of the router's other links on the same wavelength. Terminal users are connected to WDM routers. Data emerged at a WDM router from the terminal users can be sent out through any of its links in any wavelength channels. When a signal is transmitted to its destination WDM router, the data is finally distributed to the terminal users.

2.2 Wavelength-Routed Demands

In a wavelength-routed WDM network, point-to-point connections are used as communication channels to accommodate the demands. In a WDM network without wavelength conversion capability, when a demand between two nodes arrives, a connection using the same wavelength over all the links is used to establish a communication channel. In such case, the connection has to use the same wavelength along its path. This is the wavelength-continuity constraint. This constraint in WDM networks makes their resources allocation problem different in other networks.

The wavelength continuity constraint restricts the performance of the WDM networks. For example, as shown in Figure 1, the two adjacent links both have an available wavelength but the two wavelengths are not the same. Thus, another connection that goes through the two links cannot be established. Here a solid line means a busy wavelength channel while a dash line means a free wavelength channel. The numbers on the lines indicate the wavelength. To relax the constraint, wavelength converters are introduced. A node equipped with wavelength converters can convert a signal from one wavelength to another wavelength. Consider the example shown in Figure 1, if the node between the two links is equipped with a wavelength converter, the connection now can be established by using different wavelengths in the two links; the signal is converted from one wavelength to another wavelength in the intermediate node, as is shown in Figure 2. A connection using the same wavelength only is called a lightpath. A connection using more than one wavelength is called a semi-lightpath.

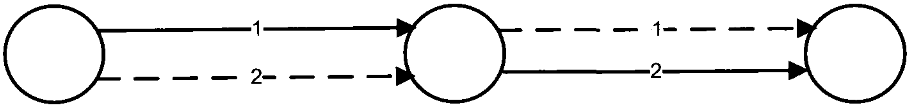

Figure 1: An example of no wavelength converter

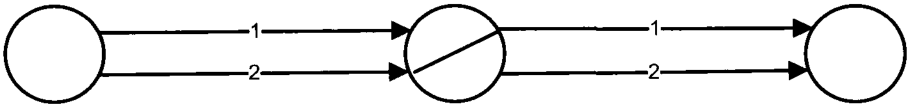

Figure 2: An example of wavelength converter

If all the nodes are equipped with sufficient wavelength converters, then the wavelength continuity constraint no longer exists. In such a case, the WDM network is equivalent to a circuit-switched network, the properties of which have been extensively studied. However, currently the wavelength conversion devices are very expensive. It is not economical to install sufficient wavelength converters at all the nodes of a WDM network. It is reasonable to assume that the WDM networks only have partial wavelength convertibility. One possible assumption, often made in the literature, is that in the WDM network some nodes have sufficient wavelength converters while other nodes have no converters. Another assumption is that in the WDM network all the nodes have a limited number of wavelength converters.

In some papers the connections are assumed to be bi-directional. In such a case, a demand is accommodated by two communication channels going in different directions while using the same wavelength on the same links. The two end nodes of a bi-directional connection can both send and receive data. In our work, we assume the connection to be one-directional. In such a case, the connection is established between a source node and a destination node. Data can only be transmitted from the source node to the destination node.

There are two reasons why we assume that connections are one-directional. The first reason is that it is more general than the bi-directional connection assumption. In a bi-directional connection between two nodes A and B, the two one-directional communication channels must be symmetrical, which means they must use the same route and wavelength in both directions. On the other hand, if there are two one-directional connections between the two nodes (one is from A to B, the other is from B to A), the two communication channels may use different routes. From this we can see that the bi-directional connection assumption is just a special case of the one-directional connection assumption. The second reason is that a network that allows one-directional connections may perform better than the same network that only allows bi-directional connections. An example is shown in Figure 3. Here, if the connections are bi-directional (as assumed in the figure), it is impossible to establish connections between AB, BC and AC at the same time. This is due to the wavelength continuity constraint and another constraint that in a bi-directional connection the two one-directional communication channels have to be symmetrical. But if the connections are one-directional, there is a way to establish such connections, as shown in Figure 4.

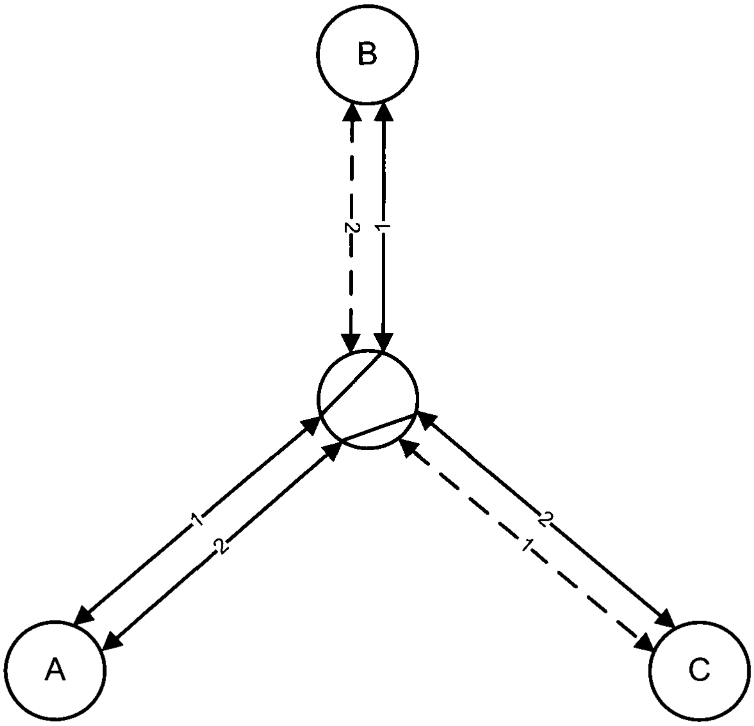

Figure 3: An example of bi-directional links

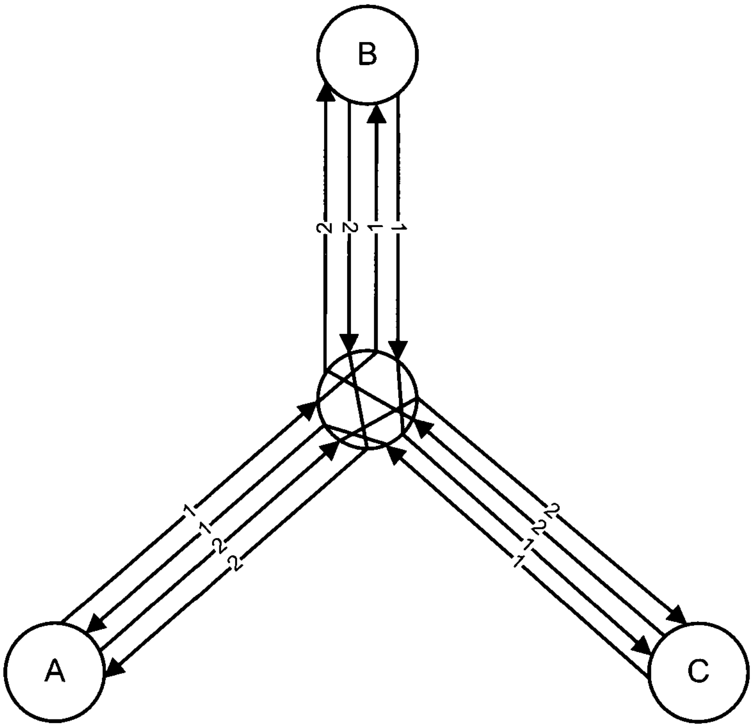

Figure 4: An example of one-directional links

2.3 Routing and Wavelength Assignment Problem

In a wavelength-routed WDM network, the connection channel is determined by both the path it traverses and the wavelength it uses. To accommodate a demand, a path with a common free wavelength along all the links has to be found. This is the called the Routing and Wavelength Assignment (RWA) problem, since the problem can be decomposed into a routing sub-problem and a wavelength assignment sub-problem. However, these sub-problems are in general not independent.

RWA schemes can be divided into two classes, one-stage and two-stage. One-stage schemes solve the routing problem and the wavelength assignment problem jointly. In such schemes, the network is often transformed into an auxiliary graph. All the routing and wavelength assignment solutions between two nodes are represented by paths between the two nodes in the auxiliary graph. Then the RWA problem is solved by finding a shortest path in the auxiliary graph. These schemes are very computationally expensive. This is because the number of possible routing-wavelength assignment combinations is very big, and the number grows dramatically as the number of nodes and wavelengths in the network increase. One-stage schemes also require link status information of the entire network, making them hard to implement.

Two-stage schemes solve the two problems separately. In these schemes, a path is determined first, and then the wavelength assignment is determined accordingly. Compared to one-stage schemes, the two-stage schemes have much less computational complexity, since they only search a part of the entire solution space. Two-stage schemes only require link status information along one or several routes, thus they can be implemented easily. However, two-stage schemes often perform worse than one-stage schemes, because they explore only a part of the solution space.

In the two-stage scheme, the two sub-problems can be solved in many ways. In the literature, many routing schemes are proposed [3]:

- Fixed Routing: only one fixed path is used for each node pair.
- Alternative Routing: a set of fixed paths is used for each node pair.
- Adaptive Routing: the path used for each node pair is determined on-the-fly, taking the current network status into account.

Fixed and Alternative Routings both use pre-computed paths. They have less computation cost and only require link status information along these paths. Adaptive Routing computes the paths dynamically. It has a higher computation cost and may need link status information of the entire network. And its performance is often better than Fixed Routing and Alternative Routings. In the thesis we use all the three schemes.

Once a path is chosen and there are several available wavelengths along the path, a scheme is needed to choose a wavelength. Many wavelength assignment schemes are proposed [3]:

- First-Fit: Choose the wavelength with the lowest index
- Random: Choose the wavelength randomly.
- Most-Used: Choose the wavelength that is most used in the current network
- Least-Used: Choose the wavelength that is least used in the current network
- Max-Sum: Choose the wavelength that would block the least number of potential demands

First-Fit is the simplest scheme and it can be implemented easily with the knowledge of the link status information along the path only. Compared with the Random Scheme, its computation cost is lower. This is because in the Random scheme, all the wavelengths must be checked before one is chosen randomly. In the First-Fit scheme, the search for an available wavelength starts at the lowest index wavelength; once a wavelength is found, the search terminates. The First-Fit scheme performs well and is often used [3]. For these reasons we use First-Fit scheme in this thesis.

The Most-Used scheme slightly outperforms the First-Fit scheme while the Least-Used scheme performs worse than the First-Fit scheme. The Max-Sum scheme has the best performance among all the schemes. However, the Most-Used, Least-Used and Max-Sum schemes all require link status information of the entire network and have very high computation costs. These schemes are not recommended for real implementations [3].

2.4 Static and Dynamic RWA Problems

One usually distinguishes between static and dynamic RWA problems. In static RWA, all the traffic demands for a given network are known in advance, and the traffic is assumed to stay in the network for very long time. With the knowledge of all the traffic demands and the network topology, the allocation problem can be formulated as an optimization problem. The objective function of the optimization problem is usually the number of traffic demands accommodated, which should be maximized. The optimization problem is proven to be NP-complete and is hard to solve. For large networks, getting the optimal solution is often not possible. Thus, some mathematical approaches are used to get near-optimal solutions.

In dynamic RWA, demands arrive one after the other and stay in the network for an unknown duration before they are released. Each time when a demand comes, the decision of allocating this demand has to be made without knowing the future demands. And it is assumed here that the demands already accommodated in the network cannot be rerouted. This behavior makes the dynamic RWA a stochastic process. Each demand is either accommodated or blocked, depending on the current status of the network. In dynamic RWA, the objective is to minimize the long-term blocking probability. Dynamic RWA is the topic we study in the thesis. We choose this topic because in the real world the demands are usually dynamic.

Chapter 3

The Multiplier Guide for Dynamic RWA

3.1 The Use of Future Traffic Information to Guide Dynamic RWA

We consider that the distribution of the dynamic traffic demands is very important for solving the dynamic RWA problem. We assume that the traffic demand is different for different source-destination pairs. Most of the existing studies of RWA schemes assume that the traffic demands are uniformly distributed among all pairs of the nodes in the network. This is a weak assumption, since in the real networks the data transmission demands arising in different nodes may not be always the same. We assume that the distribution of the dynamic traffic demands can be arbitrary, but can be predicted; and we use a matrix to represent the expected time-average distribution. Most of the existing dynamic RWA schemes do not use the knowledge of this distribution [4-6]. These schemes use the hop count as link cost and choose paths having the minimal number of hops to accommodate the dynamic demands. However, if the paths are chosen without the consideration of the distribution of dynamic demands, they may not be optimally planned. Unlike the existing schemes, we consider the distribution very useful information and always accommodate the dynamic traffic demands accordingly.

The main idea of our approach is that if the distribution of the dynamic traffic demands is known, this information can be used to guide dynamic RWA schemes and improve their performance. When a network is under a certain distribution of dynamic traffic demands, with an RWA scheme in use, some of its links may become highly utilized while others barely utilized. The utilization status of the links is very important information for dynamic RWA. Without the knowledge of this status, the dynamic RWA schemes may only try to accommodate the demands by using paths that are physically short, but may contain highly utilized links. Thus, these schemes may lead to high blocking probabilities for these dynamic demands. When the link status is considered, we can propose better dynamic RWA schemes that can balance the traffic and improve the performance.

3.2 The Multiplier Guide

In our approach, with the network topology and distribution of dynamic demands known, we compute a cost for each link to substitute the hop count. The mathematical approach used to compute these link costs has been proposed in our group previously [1,7], and it is briefly introduced in the Appendix.

In this work, the static RWA problem is considered and the cost for each link is computed in this context. We use the network topology and a traffic matrix, which is equivalent to the distribution of the dynamic demands, to formulate a static RWA problem. In this problem we want to minimize the objective function, which is the sum of all the demand rejection penalties and the channel usage costs of accepted demands, under several constraints, including the wavelength channel capacity constraints.

To solve the problem, the Lagrangian Relaxation method is used, which means that some constraints are multiplied by a number, called Lagrange Multiplier, and added to the objective function. Thus, a dual problem that has both, all the variables of the static RWA problem and the Lagrange Multipliers as new variables, is formulated. Then the dual problem is solved iteratively. Once this solution is found, we get from it, a Lagrange Multiplier for the capacity constraint of each directional link. Finally, the Lagrange Multipliers are used as factors of the link costs to solve the static RWA problem.

Mathematically, the Lagrange Multiplier is used to solve optimization problems with constraints. The value of the Lagrange Multiplier indicates how much the objective function can be improved if the associated constraint is modified [8]. In our case, the value of the Lagrange Multiplier shows how much the objective function can be improved if the capacity of the link it is associated with is increased by one unit. It is intuitively clear that the value of the Lagrange Multiplier is related to the link utilization. This relationship is further studied in Section 5.4.3.

In the following, the sum of the channel usage cost (as set in the objective function of the static RWA problem) and the Lagrange Multiplier associated with the link is used as the cost of the link for the dynamic RWA problem. We call this value the Multiplier Guide for the link. According to [1,,7], when minimal cost paths computed from the Multiplier Guide are used to accommodate the loaded demands, the static RWA problem is optimized.

The static RWA problem is formulated according to the dynamic RWA problem that we want to solve, so the Multiplier Guide obtained from the static RWA problem can help us in solving the dynamic RWA problem. We expect that, in the dynamic case, when the Multiplier Guide is used as link cost instead of the hop count to compute paths, the network performance will be improved. In the following chapters, we apply the Multiplier Guide to several existing dynamic RWA schemes and see how the performance is improved.

3.3 Examples of the Multiplier Guide

Here several examples of the Multiplier Guides generated for different scenarios are shown. In each scenario, we first present the network topology and the traffic matrix, which are used to formulate the static RWA problem. Then we show the Multiplier Guide generated for each link from the static RWA problem, which will be used as link cost for the dynamic RWA problem. We show and illustrate the examples here and use the Multiplier Guides obtained from these examples to run simulations in the following chapters.

3.3.1 The 14-Node NSFNET Network

The network architecture shown in Figure 5 is the most common topology used in the literature [1,4,16]. We assume that each link is a bi-directional fiber and has 16 wavelengths.

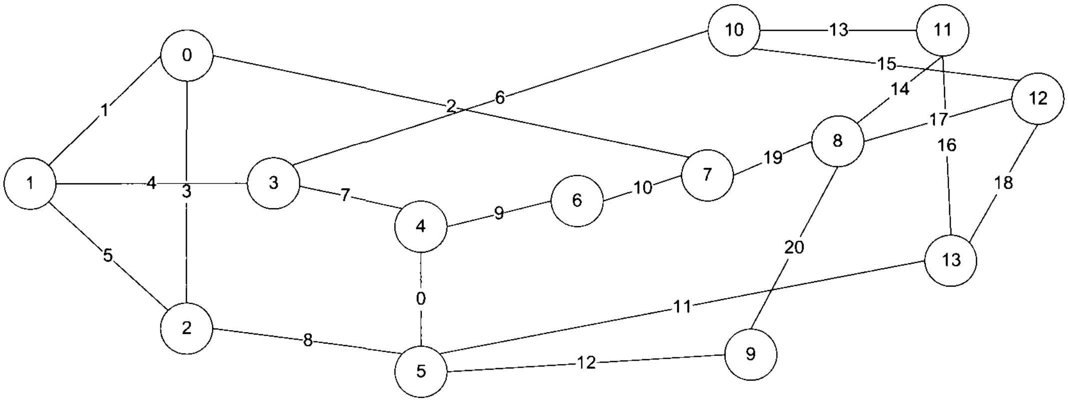

Figure 5: The 14-node NSFNET network

The assumed traffic demand matrix is shown in Table 1. The matrix element in row i and column j of the matrix represents the number of connection requests from nodes i to j. This matrix is randomly generated with each non-main diagonal element having the same probability of being 0, 1, 2 and 3.

In the static case we use a very high load of demands to compute the Multiplier Guide. This is because the Multiplier Guide indicates how critical the network resource is, and this criticality can only be seen when the network is highly loaded. Once we obtain the Multiplier Guide from the static case, we apply it to the dynamic case for different traffic loads. However, we assume that the distribution of the traffic load over the different source-destination pairs is always proportional to the values contained in the traffic matrix.

0	1	3	1	3	1	3	0	2	0	3	2	0	3
0	0	0	2	0	1	0	0	1	0	1	0	0	3
3	0	0	3	0	1	2	3	2	3	1	2	3	0
3	1	0	0	1	1	2	3	2	2	3	2	0	3
1	0	1	2	0	3	3	2	0	3	3	1	1	3
1	2	1	3	2	0	1	3	3	1	2	1	0	2
3	1	2	3	3	3	0	3	3	1	3	2	3	3
0	0	0	0	1	0	3	0	0	1	3	0	2	0
3	0	1	3	3	3	3	0	0	2	3	1	1	2
0	0	0	1	2	0	2	0	1	0	1	0	0	3
1	0	0	2	0	3	1	1	0	3	0	3	0	3
2	3	1	1	3	2	3	2	2	2	3	0	1	3
2	0	0	0	0	1	2	0	3	0	2	0	0	3
1	3	0	2	3	2	3	3	1	2	3	3	3	0

Table 1: The first traffic load matrix for the 14-node NSFNET network

The optimized Lagrange Multiplier values obtained from static RWA problem are shown in Table 2. In the objective function of the static RWA problem, the demand rejection penalty cost was 100 while the channel usage cost was 25, which is the cost per wavelength channel.

Trans. Node	Rece. Node	Multiplier Guide Value	Trans. Node	Rece. Node	Multiplier Guide Value
0	1	0.01	7	0	0.02
0	7	22.19	7	6	0. 06
0	2	0.01	7	8	17. 33
1	0	0.00	8	11	0. 01
1	3	0.04	8	12	0. 02
1	2	0.00	8	7	25. 04
2	0	0.05	8	9	12. 64
2	1	0.01	9	5	0. 04
2	5	6. 87	9	8	0. 01
3	1	0.04	10	3	21.86
3	10	31.28	10	11	0. 01
3	4	0. 06	10	12	0. 02
4	5	21.77	11	10	1. 01
4	3	4. 06	11	8	0. 03
4	6	0. 07	11	13	0. 03
5	4	16.60	12	10	0. 98
5	2	0. 06	12	8	0. 02
5	13	17. 45	12	13	0. 02
5	9	12. 69	13	5	24. 55
6	4	0. 06	13	11	0. 01
6	7	21. 90	13	12	0. 02

Table 2: The Lagrange Multiplier values generated from the first traffic load matrix

For the same topology, under different traffic loads, the Multiplier Guide values should be different. To verify this, we use the traffic load shown in Table 3, and get the Multiplier Guide values shown in Table 4. It can be seen the values are different from that in Table 2.

0	0	2	1	3	2	0	3	2	2	1	0	2	1
1	0	3	3	1	0	2	1	0	2	0	1	0	1
0	3	0	2	0	2	3	0	3	2	1	0	2	1
2	2	0	0	0	3	2	0	1	3	0	2	2	0
0	3	2	1	0	3	2	3	2	0	3	0	2	3
1	1	0	3	2	0	3	0	1	2	3	2	0	3
0	2	1	2	0	3	0	1	0	3	2	0	3	0
1	3	1	0	2	2	3	0	0	1	3	0	2	2
1	3	0	3	3	0	1	0	0	1	3	0	1	2
0	2	3	0	1	1	3	0	2	0	1	3	0	3
3	1	2	0	2	0	1	3	0	3	0	1	0	0
0	3	0	2	3	0	2	0	1	0	3	0	2	3
0	2	1	0	3	2	0	3	0	1	3	3	0	3
1	3	1	0	2	1	3	0	2	3	0	2	1	0

Table 3: The second traffic load matrix for the 14-node NSFNET network

The new Multiplier Guide values are shown in Table 4. It can be seen the values are different from that in Table 3.

Trans. Node	Rece. Node	Multiplier Guide Value	Trans. Node	Rece. Node	Multiplier Guide Value
0	1	0.03	7	0	0.03
0	2	0.03	7	6	0.03
0	7	15.34	7	8	9.39
1	0	0.01	8	7	22.07
1	2	0.02	8	9	5.21
1	3	0.07	8	11	0.04
2	0	0.01	8	12	0.04
2	1	0.04	9	5	0.08
2	5	6.46	9	8	0.03
3	1	1.11	10	3	23.11
3	4	1.18	10	11	0.02
3	10	23.13	10	12	0.01
4	3	1.06	11	8	0.02
4	5	17.98	11	10	0.05
4	6	0.41	11	13	0.02
5	2	0.04	12	8	0.03
5	4	14.73	12	10	0.05
5	9	8.60	12	13	0.03
5	13	18.75	13	5	21.54
6	4	0.03	13	11	0.02
6	7	13.65	13	12	0.02

Table 4: The Lagrange Multiplier values generated from the second traffic load matrix

3.3.2 The 28-Node Pan-European Network

Then we consider a second network topology, which is shown in Figure 6. In this topology every link is a bi-directional fiber and has 16 wavelengths. The traffic load matrix is shown in Table 5, and the Lagrange Multiplier values generated from this scenario are shown in Table 6. Here the channel usage cost was 20.

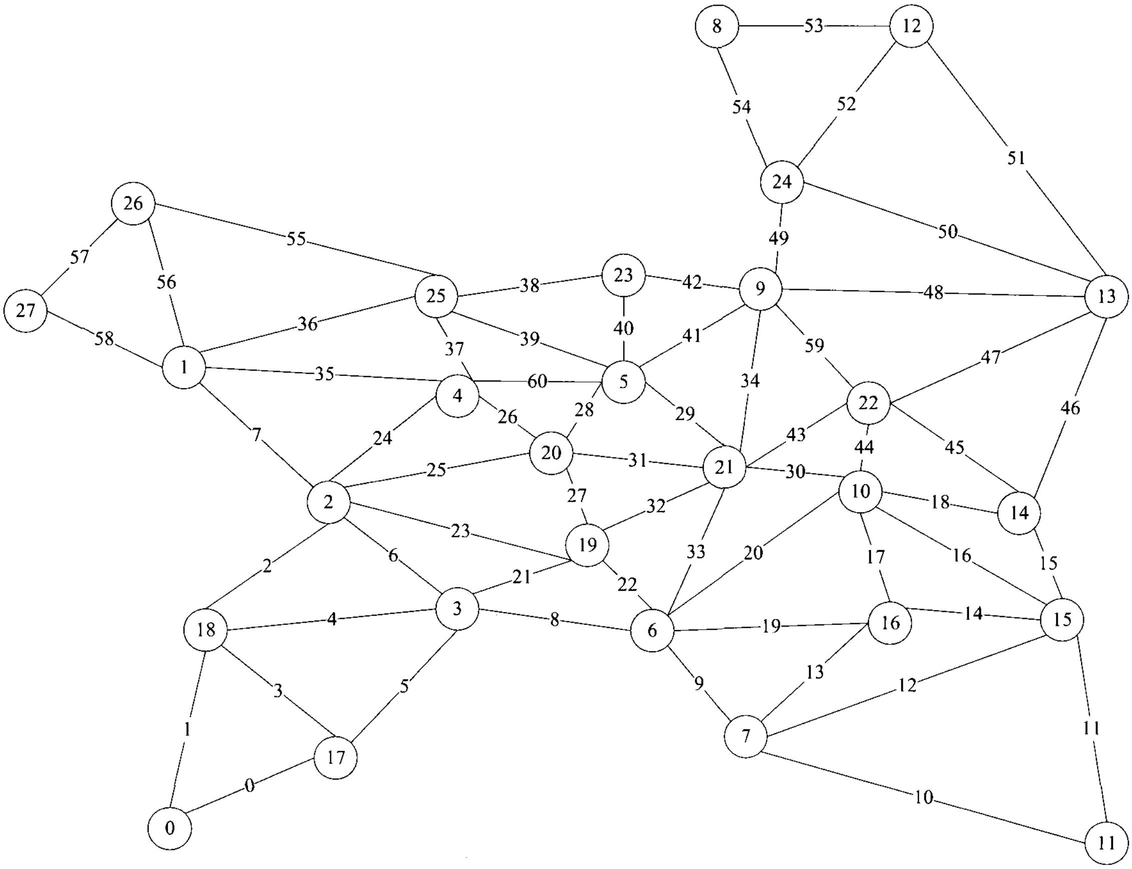

Figure 6: The 28-node Pan-European network

0	2	2	2	2	1	2	1	0	0	1	1	0	0	0	0	0	0	0	2	2	1	0	0	0	0	0	0
2	0	2	1	2	2	0	0	0	1	2	0	0	0	0	0	1	2	2	1	1	0	0	0	0	2	0	2
2	2	0	1	2	2	1	0	0	0	1	1	0	0	0	0	1	2	2	2	2	1	0	0	2	0	1	1
2	1	2	0	1	0	2	2	0	1	2	1	0	0	0	1	2	2	1	2	1	1	1	0	1	1	0	0
2	2	2	1	0	2	1	0	1	2	1	0	1	0	0	0	1	0	0	2	2	2	0	2	0	2	0	1
0	2	1	0	2	0	0	2	2	1	2	0	1	1	2	2	0	0	1	0	0	2	1	0	2	0	2	2
2	1	1	0	1	2	0	0	0	0	1	1	0	0	2	0	0	2	2	0	0	0	0	0	0	1	0	1
1	1	1	2	0	0	0	0	0	2	0	0	0	0	0	2	0	1	2	2	0	2	0	0	2	1	1	1
0	0	0	0	0	2	0	0	0	2	1	0	2	2	1	1	0	0	0	1	1	2	2	2	0	1	2	2
0	0	1	0	2	2	1	0	2	0	0	0	2	2	1	0	1	0	1	0	1	2	2	1	0	0	2	2
0	1	0	1	0	1	0	0	1	2	0	2	1	1	2	1	0	0	0	0	2	1	1	1	0	2	0	0
0	1	0	0	0	1	2	1	0	1	1	0	0	0	1	0	2	0	0	1	0	0	0	2	0	1	0	2
0	0	0	1	0	2	1	0	0	2	1	0	0	2	2	1	1	0	0	0	1	2	2	2	1	2	0	0
0	0	0	1	0	0	0	2	2	0	0	0	2	0	0	1	0	0	0	0	0	1	2	2	2	1	0	0
0	0	0	0	0	0	1	2	1	1	0	1	2	1	0	0	1	0	0	1	1	2	2	0	2	0	1	0
0	0	1	1	1	0	0	2	1	0	0	1	1	2	0	0	0	0	0	0	0	2	0	1	1	1	0	0
1	0	0	1	1	1	0	1	0	2	1	2	0	0	2	1	0	0	0	2	0	1	1	0	1	0	0	2
0	2	2	2	2	1	2	1	0	0	0	0	0	0	0	0	0	0	2	2	2	0	0	0	0	1	0	0
0	1	2	2	1	0	2	1	0	0	0	0	0	0	0	0	1	2	0	2	2	2	0	0	0	1	0	1
2	1	2	1	2	2	0	0	0	2	0	2	0	0	0	1	0	2	1	0	2	1	0	0	2	0	2	0
2	1	1	0	2	0	1	0	1	0	0	1	1	0	0	2	1	2	2	0	0	1	0	2	0	0	2	0
0	0	0	1	1	0	1	0	2	0	2	0	1	1	2	1	1	1	1	0	0	0	2	0	2	0	0	2
0	2	1	0	0	2	0	2	2	0	2	0	2	2	1	0	2	0	0	1	0	1	0	2	0	2	0	1
0	2	1	0	0	1	0	2	2	1	0	0	2	0	0	2	2	0	0	0	2	0	0	0	2	0	1	0
0	0	0	1	1	0	1	0	0	2	2	0	0	2	2	0	1	0	0	0	1	2	2	2	0	0	0	0
1	2	0	0	2	2	0	0	1	1	0	1	1	0	1	1	1	1	1	0	1	0	0	2	2	0	2	0
1	2	2	1	1	0	2	0	0	0	1	0	0	2	1	0	1	1	2	1	1	1	0	0	1	2	0	2
1	2	2	0	0	2	1	2	0	0	0	1	0	0	0	1	0	1	1	2	0	0	1	2	0	1	0	0

Table 5: The traffic load matrix for the 28-node Pan-European network

Trans Node	Rece Node	Multiplier Guide Value	Trans Node	Rece Node	Multiplier Guide Value	Trans Node	Rece Node	Multiplier Guide Value
0	17	0 00	9	21	0 38	18	3	2 77
0	18	1 06	9	5	2 29	19	3	0 05
1	2	16 59	9	23	2 71	19	6	0 06
1	4	0 03	9	13	0 04	19	2	0 37
1	25	0 01	9	24	14 45	19	20	0 02
1	26	0 00	9	22	0 04	19	21	0 05
1	27	0 04	10	15	0 03	20	2	0 03
2	18	10 40	10	16	0 02	20	4	0 42
2	3	0 03	10	14	0 02	20	19	0 01
2	1	8 67	10	6	0 01	20	5	0 02
2	19	0 06	10	21	0 88	20	21	0 05
2	4	0 05	10	22	0 02	21	5	1 87
2	20	0 06	11	7	0 01	21	10	1 94
3	18	0 08	11	15	0 01	21	20	0 36
3	17	3 50	12	13	10 92	21	19	0 02
3	2	0 06	12	24	0 00	21	6	0 04
3	6	10 49	12	8	0 00	21	9	0 08
3	19	0 12	13	14	0 08	21	22	0 02
4	2	0 02	13	22	0 05	22	21	0 38
4	20	0 05	13	9	0 05	22	10	0 03
4	1	0 30	13	24	0 07	22	14	0 02
4	25	0 01	13	12	2 76	22	13	0 04
4	5	0 03	14	15	0 04	22	9	0 02
5	20	0 01	14	10	0 02	23	25	0 01
5	21	3 56	14	22	0 01	23	5	0 01
5	25	0 40	14	13	0 05	23	9	3 39
5	23	0 01	15	11	0 01	24	9	12 20
5	9	3 34	15	7	0 01	24	13	0 86
5	4	0 03	15	16	0 00	24	12	0 00
6	3	1 28	15	14	0 04	24	8	2 57
6	7	1 94	15	10	0 03	25	1	0 00
6	16	0 04	16	7	0 00	25	4	0 01
6	10	0 01	16	15	0 01	25	23	0 03
6	19	0 36	16	10	0 02	25	5	0 06
6	21	0 03	16	6	0 02	25	26	0 03
7	6	8 79	17	0	0 00	26	25	0 05
7	11	0 01	17	18	0 00	26	1	0 03
7	15	0 00	17	3	3 90	26	27	0 00
7	16	0 00	18	0	3 35	27	26	0 00
8	12	0 00	18	2	4 02	27	1	0 03
8	24	7 55	18	17	0 00			

Table 6: The Lagrange Multiplier values generated from the traffic load matrix

Chapter 4

Simulation Framework

4.1 Discrete Event Simulation

We want to simulate the resource allocation process in WDM networks under dynamic traffic demands. In this process the dynamic demands arrive and leave randomly as time goes by. It is assumed that demands arrive with the same probability at any time randomly. It can be proved that if the demands arrive in such a pattern, the inter-arrival time of the demands is exponentially distributed. An incoming demand is either accepted or rejected, depending on the current network status and the RWA scheme used. Once accepted, a demand stays in the network for a random duration and then is released. It is assumed this service time is exponentially distributed.

We use the discrete event simulation method to simulate this dynamic process. During this process, the network status only changes when the arrival or the departure of a demand happens. These arrivals and departures are called events. In the discrete event simulation method, the dynamic process is not simulated continuously. Instead, a sequence of events is used to represent the dynamic process. We only simulate the times when events happen. Each time an event happens, the current system time is updated, statistics about the network for the period from the previous event to the current event is collected, and finally the changes of the system status due to this event are made.

In a discrete event simulation, a future event list is used to manage the future events. This list stores all known future events and sorts them by their time. After an event is handled, the next event in the future event list is considered. An event may schedule other events that happen in the future and these events are inserted into the future event list. When a termination condition is satisfied, the simulation stops.

The advantage of using discrete event simulation method rather than continuous simulation is that it is more efficient. In a continuous simulation, the system time is simulated by using many time slots while events only happen in a few time slots. Thus the simulation has to handle many empty slots, leading to low efficiency. Also in a continuous simulation, the time to run a simulation is almost proportional to the simulated time duration. Its efficiency is especially low when the inter-event time intervals are large. On the other hand, in the discrete event simulation only events are simulated. The time to run a simulation is only dependent on how many events happen during the time the system is simulated.

4.2 The Events in Dynamic RWA Simulations

The WDM network under dynamic traffic that we want to simulate can be described as a queuing system with multiple servers and a rejection probability. In the discrete event simulation of such a system, three kinds of events must be considered: arrival events, departure events and observation events. The latter are used to divide the simulated time duration into equal length time intervals, so the statistics in these intervals can be collected independently; these events are not necessary for simulating the system but are useful for collecting statistics.

4.2.1 Arrival Events

The processing of an arrival event is presented in the following:

In a first step, the source and destination nodes of the incoming demand are randomly generated. The distribution of the demands follows the traffic matrix used in the static RWA problem. We use one of the three demand matrices shown in Tables 1, 3 and 5 in Section 3.3.1. Let N denote the sum of all the elements in the chosen traffic matrix. A demand has, according to the traffic matrix, the probability $\frac{A(i,j)}{N}$ to be a connection request transmitting from i to j, where $A(i,j)$ stands for the element of the traffic matrix in the ith row and jth column. It is assumed that all demands have the same bandwidth requirement, namely one wavelength channel.

During the second step, the system time is updated to be equal to the time of this event. The statistics of the interval between the previous event and the current event are updated, such as the average network utilization.

In the third step, the dynamic RWA scheme tries to accept the demand that arrives in the event. If a connection can be found, the demand is accepted. At first a network status change is made, which means the network resources used by this demand are marked used. Then the total number of accepted demands is increased. Then a departure event of this demand is scheduled. The time of the departure event is set to be the current system time plus a random time, which is the exponentially distributed service time. If a connection cannot be found, the demand is rejected. In this case the network status remains the same, but the total number of rejected demands is increased. No departure event is scheduled in this case.

In the final step, a next arrival event is scheduled. The time of the next arrival event is set to be the current system time plus a random time, which is the exponentially distributed inter-arrival time.

The flow chart for the processing of an arrival event is shown in Figure 7.

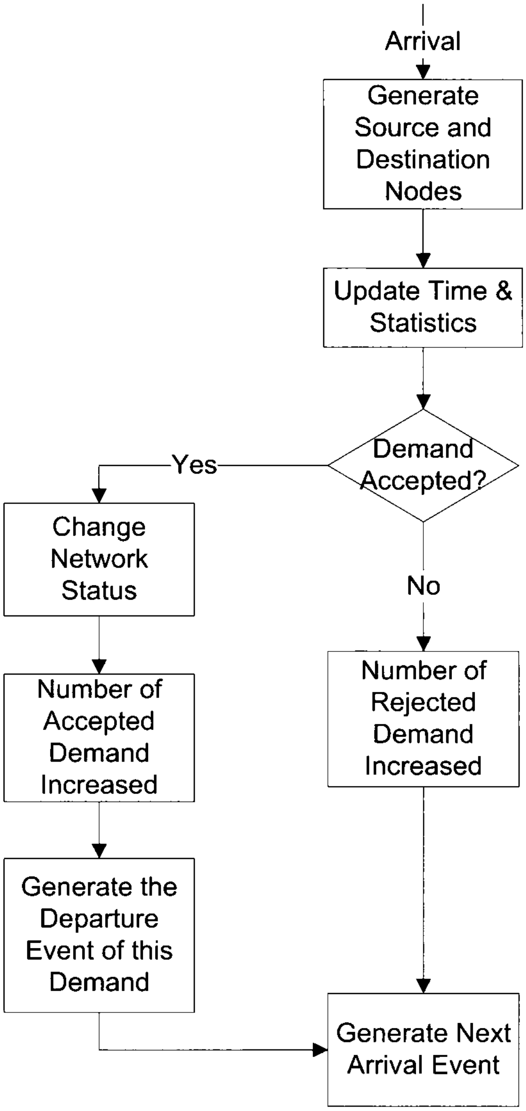

Figure 7: The flowchart of an arrival event

4.2.2 Departure Events

The processing of the departure event is presented here: At first the system time is updated to be the time of this event. The time-average statistics of the interval between the previous event and the current event are updated. Then the demand leaves the network. The network status changes as the network resources used by this demand are released. The flow chart for the processing of the event is shown in Figure 8.

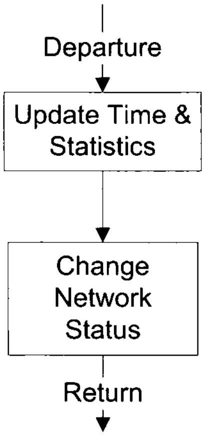

Figure 8: The flowchart of a departure event

4.2.3 Observation Events

The processing of observation events is presented in the following:

Similar to the other two events, the system time must be updated and the time-average statistics of the interval between the previous event and the current event are collected. Then the statistics of the period between this observation event and the previous observation event must be determined and stored. These statistics include the network utilization and blocking probability. Then all the statistics variables are reset. Finally, another observation event is generated. Its time is set to be the current time plus the length of the time interval between two observation events. The flowchart of the observation event is shown in Figure 9.

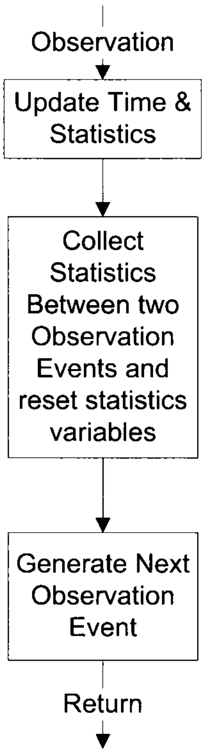

Figure 9: The flowchart of an observation event

4.3 Statistics

4.3.1 Initial State Removal

The simulation process has two periods: initial period and stationary period. At the beginning of the simulation, the network starts with zero utilization. Then the demands come in and the network is gradually filled with demands. During this period, most of the arriving demands are accepted. There are more demands coming to the network than leaving. The network utilization is growing. This transition period is called the initial period. After that, the network is in a stable state. The numbers of demands coming and leaving the network are statistically equal, and the network utilization remains at a fixed level, except for statistical fluctuations.

To get more accurate network performance results, the statistics obtained from the initial period of the simulation needs to be removed. It is necessary to figure out when the network enters the stationary period and to collect statistics only after this point in time.

The statistics obtained during the initial period are removed by using observation events. In our simulation, the entire simulated time duration is divided into 100 equal length intervals. The statistics are collected independently in each interval. We plot the network load obtained in all the intervals. The network load is the average number of demands staying in the network. From this plot, the time the network reaches stable state can be easily obtained. An example is shown in Figure 10. From the figure it can be seen that after about 6 intervals, the network reaches the stable state.

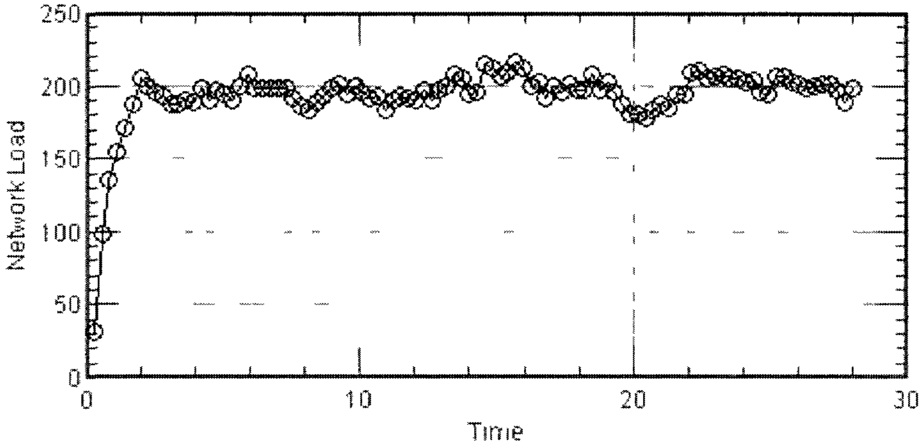

Figure 10: An example showing the initial and stable periods

After the network load over time is plotted, some initial observation intervals are removed accordingly. Finally the statistics are obtained by calculating the average over all the observation intervals that are kept.

4.3.2 Confidence Interval

Statistics generated from a single simulation is not accurate. To obtain more reliable results and an estimation of the statistical error, several replications of the same simulations, but starting with different random seeds, are performed and the results obtained from the replications are used to compute the confidence interval. In the following chapters, we run several replications for every individual simulation. The confidence interval is computed afterwards. If the 90% confidence interval is smaller than 10% of the result, we stop. Otherwise we run longer simulations or more replications until this condition is satisfied.

We show an example of the confidence interval in the following. We simulate the processes of a dynamic RWA scheme operating in a network under different network traffic loads, the details of which are explained in Section 5.1. For every given network traffic load, we do 10 replications and each replication has 1,000,000 arrival demands. The mean results and their 90% confidence intervals are shown in Figure 11.

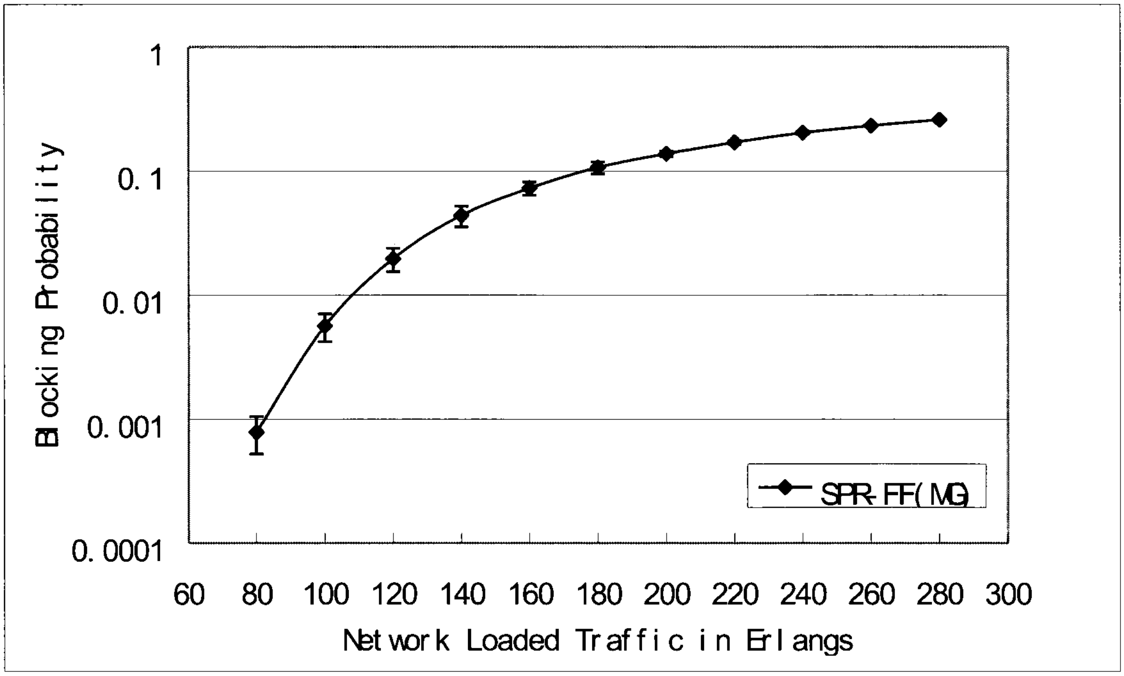

Figure 11 An example of confidence interval

It is found that, when the network loaded traffic is lower than 180 Erlangs, the 90% confidence interval becomes greater than 10%. In these cases more simulations (longer simulations or more replications) are needed to get more valid results. So in Section 5.4.1 we run more simulations for these cases until the confidence intervals are smaller than 10%.

In fact, when the confidence interval is smaller than 10%, it can hardly be seen when plotted. As a result, in the remaining parts of the thesis, we no longer plot the confidence intervals of the results.

Chapter 5

Applying Multiplier Guide to Two-Stage Schemes

In this chapter we apply the Multiplier Guide to several well-known two-stage dynamic RWA schemes. These schemes are Shortest Path Routing First-Fit (SPR-FF), Fixed Alternative Routing First-Fit (FAR-FF) and Least Loaded Routing First-Fit (LLR-FF). We run simulations to find out whether the Multiplier Guide can improve the performance of these existing schemes. We also study the performance of the schemes that set link cost by using link utilizations obtained from simulations. And we compare the performance of these schemes with the schemes that set link cost by using the Multiplier Guide.

5.1 Shortest Path Routing First-Fit Scheme

The simplest way to establish connection between a node pair is to use shortest path routing. In this case the path connecting two nodes is always the fixed shortest path. Due to its simplicity, Shortest Path Routing scheme has been studied extensively in the literature [9, 10,4].

The Shortest Path Routing First-Fit scheme solves the routing sub-problem by using a fixed route and solves the wavelength assignment sub-problem by using the First-Fit scheme. In this scheme, only one fixed path is used for accommodating demands between each node pair. When a connection between this node pair needs to be established, a common free wavelength is searched along the path. If such a common free wavelength exists, the demand is accepted; otherwise the demand is rejected. If several available wavelengths exist, the one having the lowest wavelength index is chosen.

The SPR-FF scheme can be implemented easily. Before carrying any dynamic demands, the fixed shortest path used for every node pair in the network is computed in advance.

When a dynamic demand comes, the corresponding nodes check the wavelength availability on the fixed path and try to accommodate the demand. In this scheme, only the link status information along the fixed path is needed and the computation cost is low.

5.2 Fixed Alternative Routing First-Fit and Least Loaded Routing First-Fit Schemes

In the Shortest Path Routing scheme only one path is used to carry the demands occurring between each node pair. Once a common free wavelength along the path cannot be found along the path, the demand is rejected. This scheme does not consider any alternative paths and its performance is therefore not optimal.

To improve the performance of the Shortest Path Routing scheme, the Fixed Alternative Routing [11] and Least Loaded Routing [12] schemes are proposed. To solve the routing sub-problem, both of them use a set of pre-computed paths to accommodate the demands between each node pair. In the Fixed Alternative Routing scheme, the paths used for each node pair are indexed. The scheme searches the paths one by one, starting from the lowest indexed one and terminating with highest indexed one. Once an available wavelength on a path is found, the search terminates. The wavelength found on that path is chosen to accommodate the demand. In the Least Loaded Routing scheme, all the paths have to be searched before one is chosen. After the search, the path having the largest number of available wavelengths is chosen to accommodate the demand. In both schemes, once the path is chosen and there are several available wavelengths on that path, they use the First-Fit scheme to solve the wavelength assignment sub-problem.

In the FAR-FF and LLR-FF schemes we implemented, a set of two link disjoint paths is used to accommodate the dynamic demands for each node pairs. The first path is the shortest path determined by running the shortest path algorithm. The second path is determined by running the shortest path algorithm again after the links of the first path are removed from the network.

Both of the FAR-FF and LLR-FF schemes only need link status information along a set of paths. The FAR-FF scheme can be seen as an extension of the SPR-FF scheme. Its computation cost is higher than the SPR-FF scheme because it checks the wavelength availability on multiple paths. The LLR-FF scheme has higher computation cost than FAR-FF scheme. This is because LLR-FF scheme always has to check all the paths to find the one with the most available wavelengths. On the other hand, the FAR-FF scheme terminates its search once it finds a path having an available wavelength.

5.3 Applying Multiplier Guide

In all the three two-stage schemes, the paths for all the node pairs are often pre-computed before the network is loaded with any demands. How these paths are planned is a very important factor to the network performance. It is important to figure out a way of optimally planning these paths.

The existing schemes use hop count as link cost to compute its shortest paths [4] [13]. Hop count represents the hop distance in the topology and is an important factor to be considered. However, the performance of Shortest Path Routing scheme using hop count as link cost may not be good. The shortest path may not be the best path. And finding shortest paths by using hop count does not make use of the traffic demand information. It is possible that some links end up used by many paths while other links are barely used by any paths. If this happens, the network may have congestion in those highly used links and its performance may be bad.

We use the Multiplier Guide as link cost to compute the paths for all the node pairs. The concept of Multiplier Guide is introduced in Chapter 3. It is known that the Multiplier Guide is the optimized link cost that can minimize the demand rejection penalties and channel usage costs in the static RWA problem. It is believed that the Multiplier Guide can also improve the performance in a dynamic RWA problem similar to the static RWA problem.

5.4 Simulation Results

5.4.1 The Performance Improvement of the Multiplier Guide

We run simulations to see whether the Multiplier Guide can improve the performance of the three schemes. We use the two network topologies shown in Chapter 3. In the 14-node topology, two groups of Multiplier Guides are generated from two different traffic demand matrices. In the 28-node Pan-European network topology, only one traffic demand matrix is used and only one group of Multiplier Guides is generated.

We assume the service time of the dynamic demands to be exponentially distributed with a mean time equals to one. The arrival rate, which is the average number of demands that would occur in a time unit, varies. We apply different arrival rates to simulate different network loads.

The SPR-FF, FAR-FF and LLR-FF schemes using hop count (HC) or Multiplier Guide (MG) as link cost to compute their paths are simulated separately. When hop count is used and there are multiple equal-length paths between a node pair, the tie is broken randomly. Finally the results are compared and shown in the following diagrams. The results generated from the 14-node NSFNET network with the first traffic matrix are shown in Figure 12. The results generated from the same topology with the second traffic matrix are shown in Figure 13. The results generated from the 28-node Pan-European network are shown in Figure 14. In these figures the blocking probabilities of the three schemes against network loaded traffic are plotted.

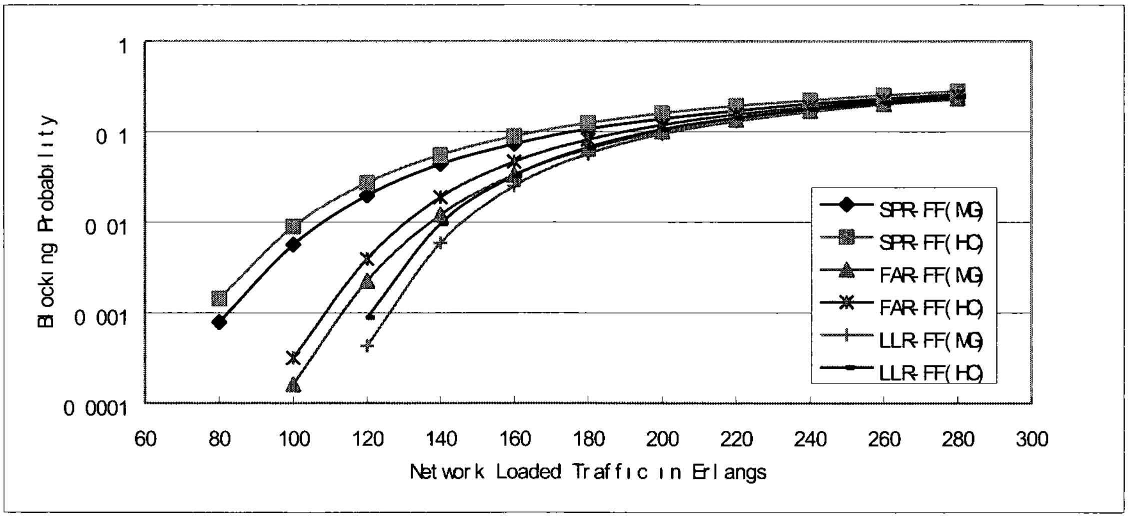

Figure 12: Blocking probabilities in the 14-node network with the first traffic matrix

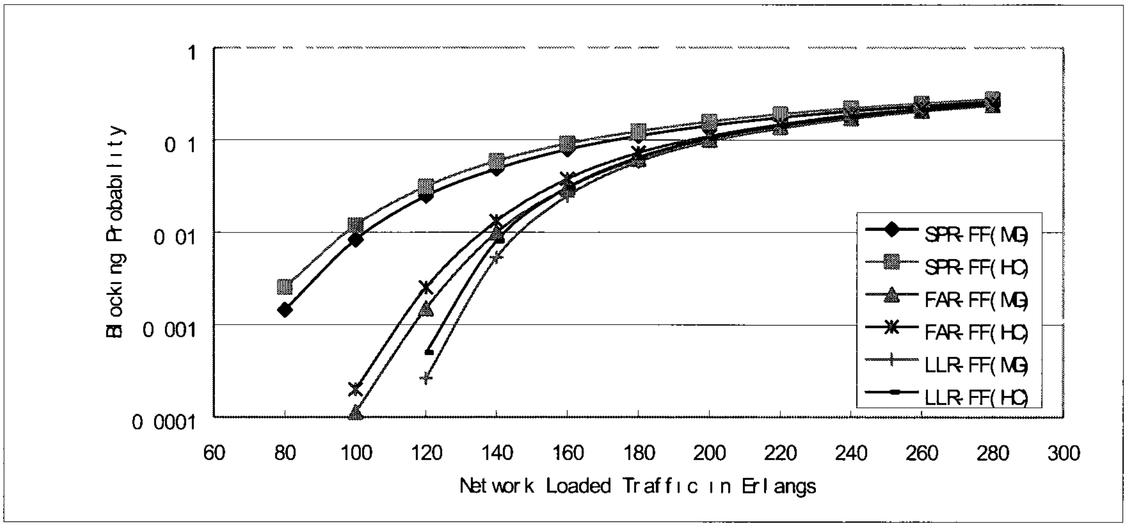

Figure 13: Blocking probabilities in the 14-node network with the second traffic matrix

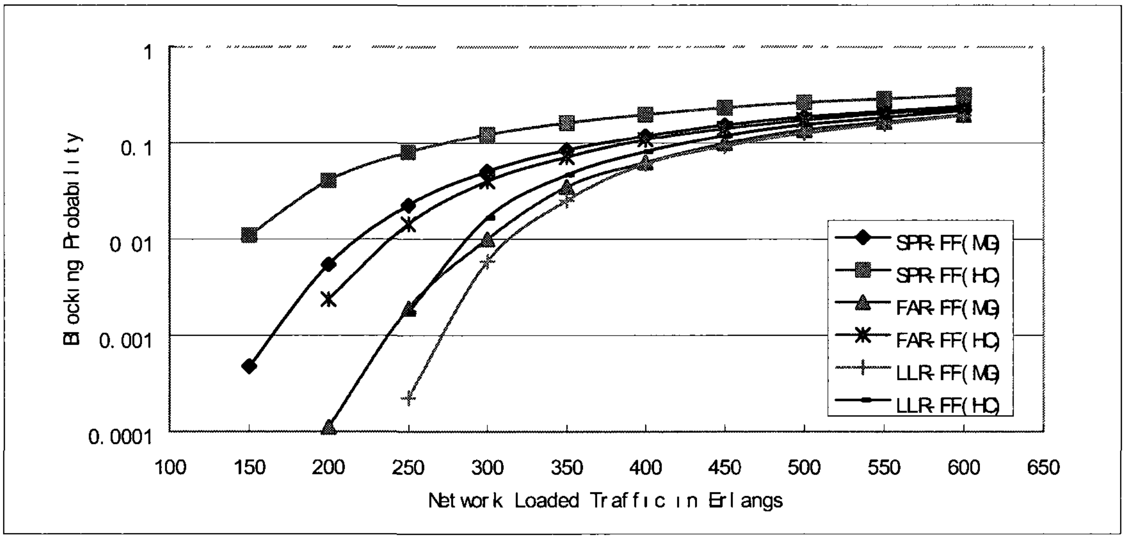

Figure 14: Blocking probabilities in the 28-node network

Several observations can be obtained from the simulations. First, it is verified in the three different circumstances that the Multiplier Guide can improve the performance of the SPR-FF, FAR-FF and LLR-FF schemes. This is a very important conclusion. Second, the improvement of using Multiplier Guide increases as the network loaded traffic decreases. Thirdly, the improvement in the second network topology is more obvious than that in the first topology.

When the SPR-FF, FAR-FF and LLR-FF schemes are compared, we see that the LLR-FF scheme has the best performance, the FAR-FF scheme performs in between and the SPR-FF scheme has the worst performance. This is because, compared with the FAR-FF scheme, the LLR-FF scheme can choose the less congested path to carry the traffic demands. And both the FAR-FF and the LLR-FF schemes use two paths for each node pair, which is better than a single path as used in the SPR-FF scheme.

5.4.2 Network Utilization

The definition of network utilization is given in the followings. Let $U(t)$denote the network utilization at time t, which is equal to the number of utilized wavelength channels at that time, in the whole network, divided by the total number of wavelength channels. Then the network utilization over simulation time period T is calculated by

$$U = \left(\int_0^T U(t)dt \right) / T$$

In the following, we show the network utilization obtained for the three schemes used in the two networks. The results of the three schemes using hop count and Multiplier Guide are shown. In Figure 15, the network utilization of the three schemes used in the 14-node NSFNET network under the dynamic traffic obeying the first traffic matrix is shown. In Figure 16, the network utilization of the three schemes used in the 28-node Pan-European network under the dynamic traffic obeying the given traffic matrix is shown.

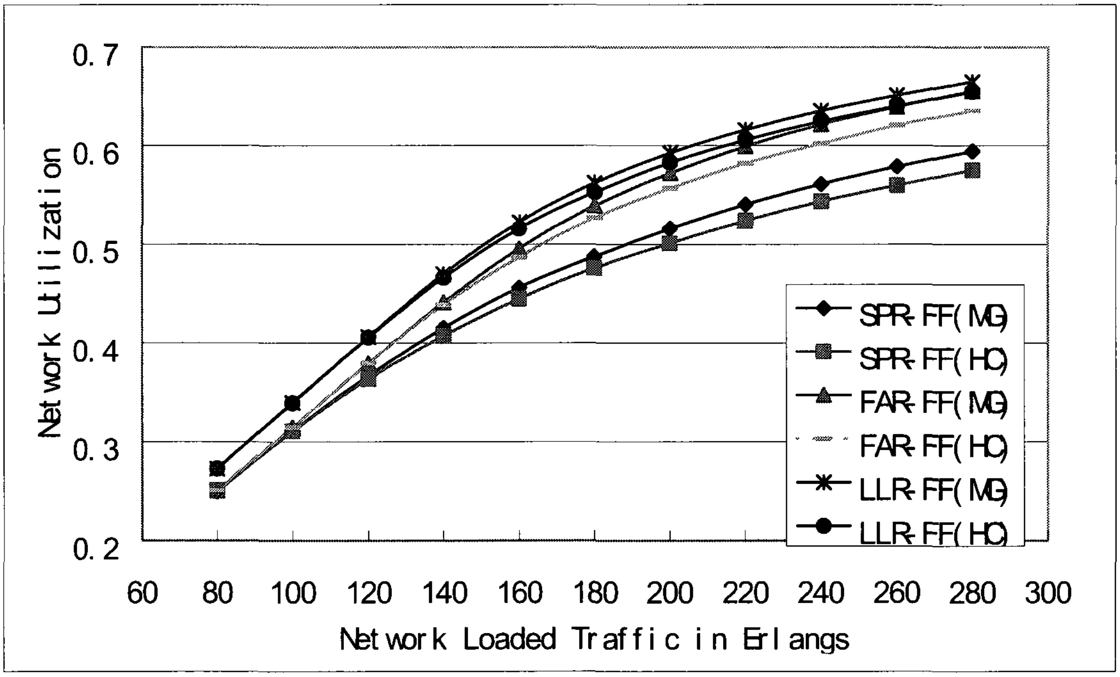

Figure 15: Network utilization in the 14-node network

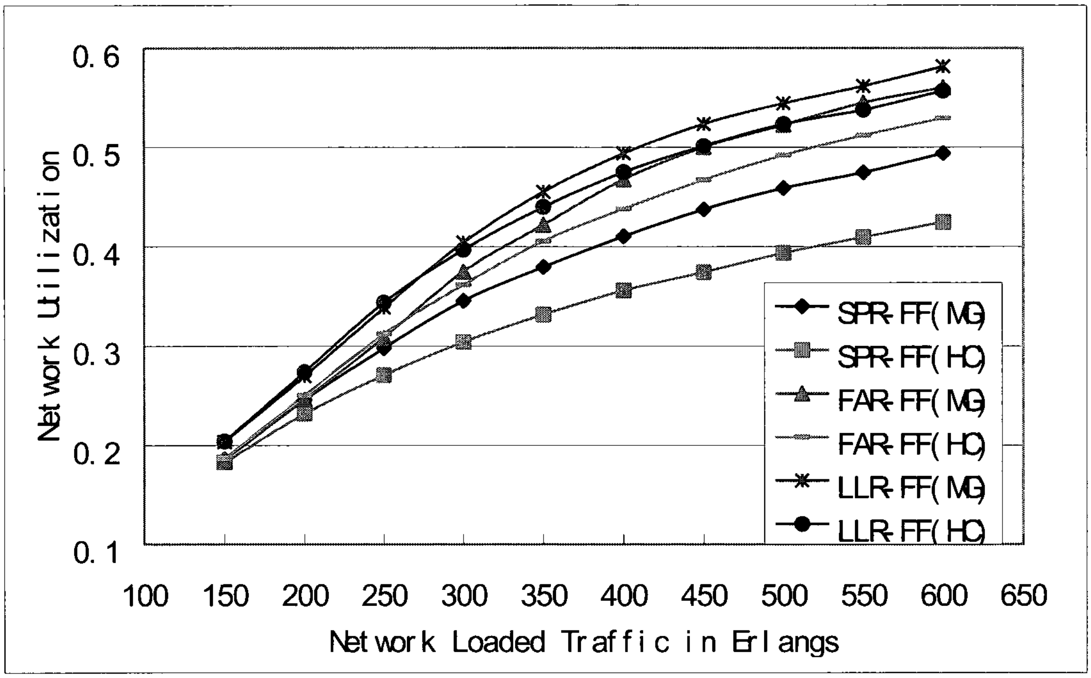

Figure 16: Network utilization in the 28-node network

From the results we see that, when different schemes are compared, the SPR-FF scheme has the lowest network utilization, the FAR-FF scheme has higher network utilization while the LLR-FF scheme has the highest network utilization. For each given scheme, we see that, under high and medium traffic loads, the network utilization is higher when the Multiplier Guide is used rather than the hop count. This difference becomes smaller and finally disappears as the traffic load goes down. Another fact is that this difference is more obvious in the 28-node network than the 14-node network.

This difference is expected, since the use of the Multiplier Guide decreases the rejection rate and therefore the load of the network increases. The reason why under low traffic loads the network utilization improvement is small is that in this case the blocking probability is very low. For both schemes, Multiplier Guide and Hop count, most of the demands are accepted. Thus the difference in network utilization is unnoticeable. When the traffic load is high, the network is congested and many demands are rejected. In this case, the difference in network utilization between the schemes using the Multiplier Guide and hop count can be easily seen.

5.4.3 Link Utilization

The definition of link utilization is given in the following. Let $U_L(t)$ denote the utilization at time t, of the directional link L, which is equal to the number of utilized wavelength channels, on that link, at that time divided by the total number of wavelength channels. Then the utilization of the directional link L over the simulation time period T is calculated by $U_L = \left(\int_0^T U_L(t) dt \right) / T$

Table 7 gives an example of the link utilizations. These link utilizations are obtained from applying the SPR-FF scheme in the 14-node NSFNET network. The hop count and the Multiplier Guide are used separately in this scheme for comparison. In both cases the network is under 120 Erlangs traffic loads. Under this traffic load the blocking probability is around 0.01. We choose this traffic load because real networks are often operating under similar traffic loads.

Trans. Node	Rece. Node	r Guide Value	Utilization (MG)	Utilization (HC)	Trans. Node	Rece. Node	r Guide Value	Utilization (MG)	Utilization (HC)
0	1	0.032	0. 21	0.23	7	0	0.034	0. 26	0.36
0	2	0.028	0. 18	0.25	7	6	0.033	0. 50	0.38
0	7	15.335	0. 41	0.49	7	8	9.394	0. 49	0.52
1	0	0.009	0. 16	0.23	8	7	22.066	0. 59	0.55
1	2	0.015	0. 10	0.10	8	9	5.207	0. 31	0.35
1	3	0.068	0. 36	0.36	8	11	0.043	0. 13	0.42
2	0	0.006	0. 34	0.34	8	12	0.041	0. 46	0.18
2	1	0.042	0. 23	0.23	9	5	0.078	0. 31	0.35
2	5	6.461	0. 48	0.47	9	8	0.028	0. 18	0.13
3	1	1.108	0. 26	0.31	10	3	23.106	0. 41	0.42
3	4	1.182	0. 41	0.46	10	11	0.017	0. 15	0.37
3	10	23.127	0. 57	0.55	10	12	0.014	0. 41	0.03
4	3	1.057	0. 48	0.54	11	8	0.022	0. 28	0.55
4	5	17.981	0. 54	0.62	11	10	0.048	0. 26	0.61
4	6	0.413	0. 44	0.40	11	13	0.022	0. 18	0.28
5	2	0.042	0. 41	0.33	12	8	0.027	0. 42	0.18
5	4	14.734	0. 44	0.63	12	10	0.051	0. 52	0.05
5	9	8.602	0. 39	0.33	12	13	0.027	0. 38	0.10
5	13	18.749	0. 60	0.62	13	5	21.54	0. 48	0.45
6	4	0.032	0. 55	0.42	13	11	0.022	0. 16	0.37
6	7	13.645	0. 49	0.48	13	12	0.018	0. 49	0.15

Table 7: Link utilization in the 14-node network

From the table we see that the links having high Multiplier Guide values often have high utilizations. However, there are also many exceptions. When the Multiplier Guide is used rather than the hop count, some previously highly used links become less congested. But there are also many exceptions. In general, from the link utilizations, nothing very clear and conclusive can be observed.

In the 28-node Pan-European network, similar results are observed. Due to the size of that network, the table of link utilizations is too big and the results are not shown here.

5.4.4 The Robustness of the Multiplier Guide

Previously we assumed that the distribution of dynamic traffic demands strictly obeys the traffic demand matrix. In the real world the distribution may change. The Multiplier Guide is generated for a given network and a given traffic demand matrix. An interesting question is how the network performance would be if the Multiplier Guide is applied to a network with its dynamic demands obeying a different traffic demand matrix. This is a test of the robustness of the Multiplier Guide. To find the answer, several new groups of simulations are made.

In the new groups of simulations, uniform distributed dynamic demands are generated. Similarly, the SPR-FF, FAR-FF and LLR-FF schemes are used to accommodate the dynamic demands. In the two network topologies, hop count and Multiplier Guide are used separately for paths planning. In the 14-node NSFNET network, we only use the Multiplier Guides generated from the first traffic demand matrix.

The simulation results are shown as follows. The results generated from the 14-node NSFNET network are shown in Figure 17. The results generated from the 28-node Pan-European network are shown in Figure 18.

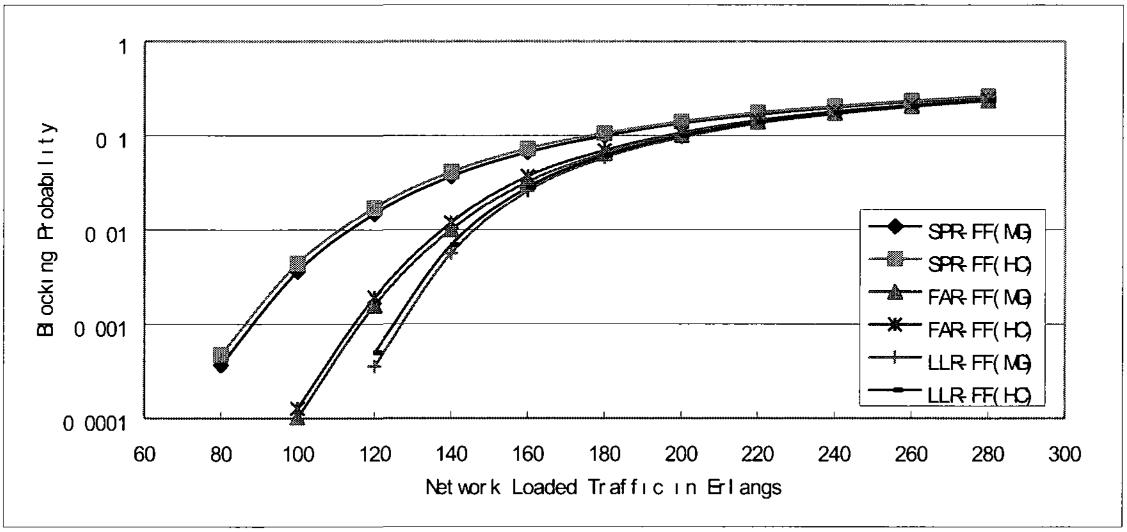

Figure 17: Blocking probabilities in the 14-node network under uniform traffic demands

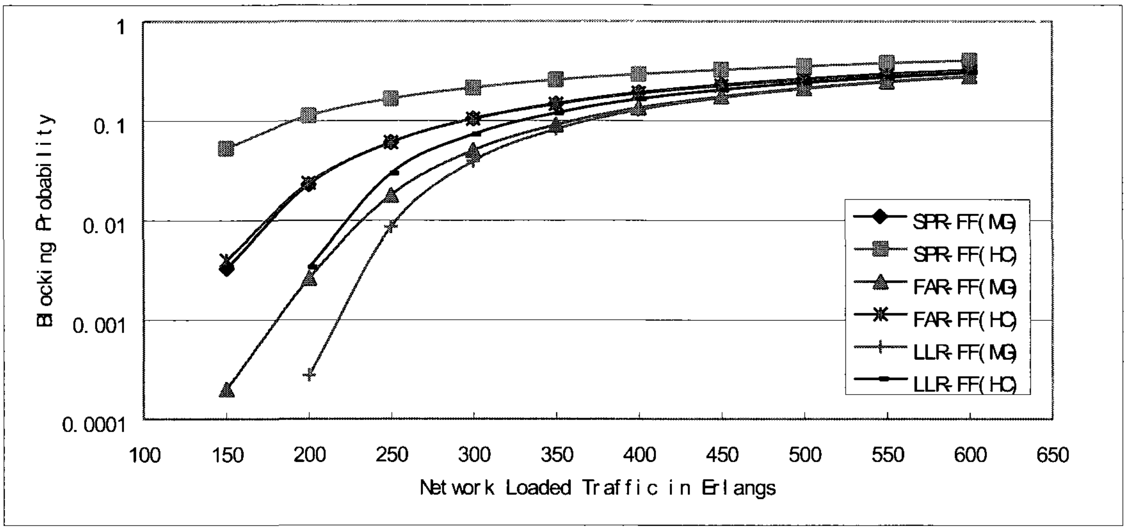

Figure 18: Blocking probabilities in the 28-node network under uniform traffic demands

It can be seen that in both topologies, when the demands do not obey the traffic demand matrix used to generate the Multiplier Guide, the improvement of using the Multiplier Guide improvement still exists but the is smaller. The improvement in the 14-node NSFNET network becomes very small. The improvement in the 28-node Pan-European network is still considerable.

The possible reason that the improvement exists but becomes smaller is because the Multiplier Guide is generated considering both network topology and a traffic demand matrix. Even if the traffic demand matrix is not obeyed, the network topology is still the same. The Multiplier Guide is still partially useful.

5.4.5 Comparison with the Other Schemes that Take Link Utilization into Consideration

The Multiplier Guide for a link is computed mathematically and is a guide indicating the importance of a link. Another way to get the importance of a link is to observe its utilization from simulations. In [14], a Best among the Shortest Routes (BSR) scheme is proposed where the link cost is determined from the link utilization obtained from simulations or measurements. We compare the performance of the BSR scheme with that of the SPR-FF schemes. In the BSR scheme, the link utilization is only used for breaking ties among equal cost paths. Therefore, we propose two other schemes that take the link utilization into account in a more important manner.

The BSR scheme is described as follows. The link cost is determined by iterative simulations. Let $c(l)_i$ and $u(l)_i$ denote the cost and utilization of link l in the i th iteration, respectively. Then, the link cost in the $(i+1)$-th iteration is determined by the formula $c(l)_{i+1} = 0.9999 * c(l)_i + 0.0001 * u(l)_i$. Initially the cost of all the links are set to be 1. After the completion of an iteration, the link costs for the new iteration are computed by applying the current link utilizations and link costs to the formula. In the BSR scheme, once the link costs are computed, the Dijkstra algorithm is used to compute one shortest path for each node pair. The First-Fit scheme is chosen for wavelength selection in the BSR scheme.

In [14], the total number of iterations was determined after a preliminary study. In that study, the number of iterations was first set to be 100. Then it was found that, in all the tested cases, the BSR scheme found its best solution of routing within 20 iterations. Although one cannot prove that the best solution can always be found within 20 iterations, the total number of iterations in the BSR scheme was chosen to be 20, which is a good compromise between finding the best solution and reasonable execution time.

We also propose two schemes, namely Using Link Utilization as Link Cost 1 and 2 (ULULC1 and ULULC2). Similar to the BSR scheme, in these two schemes, the link costs are determined by iterative simulations. In the first iteration all the links have costs equal to

1. Let $c(l)_i$ and $u(l)_i$ denote the cost and utilization of link l in the ith iteration. Let ave_i denote the average link utilization in the ith iteration. In the ULULC1 scheme, the link cost in the $(i+1)$th iteration is determined by the formula $c(l)_{i+1} = 1 + u(l)_i - ave_i$, which means that the link cost is a linearly increasing function of the link utilization. In the ULCLC2 scheme, the formula is $c(l)_{i+1} = e^{u(l)_i - ave_i}$, which means that the link cost is a nonlinearly increasing function of the link utilization. In both schemes, a single path is used to accommodate the traffic and the First-Fit scheme is used to decide the wavelength. In the ULULC1 &2 schemes, the total number of iterations is chosen to be 20, the reason is similar to that of the BSR scheme.

The simulation results of the SPR-FF (MG), BSR, ULULC1 and ULULC2 schemes are shown in the following. Figure 19 shows the results obtained from the 14-node NSFNET network under the dynamic traffic obeying the first traffic matrix. Figure 20 shows the results obtained from the 28-node Pan-European network under the traffic obeying the given traffic matrix.

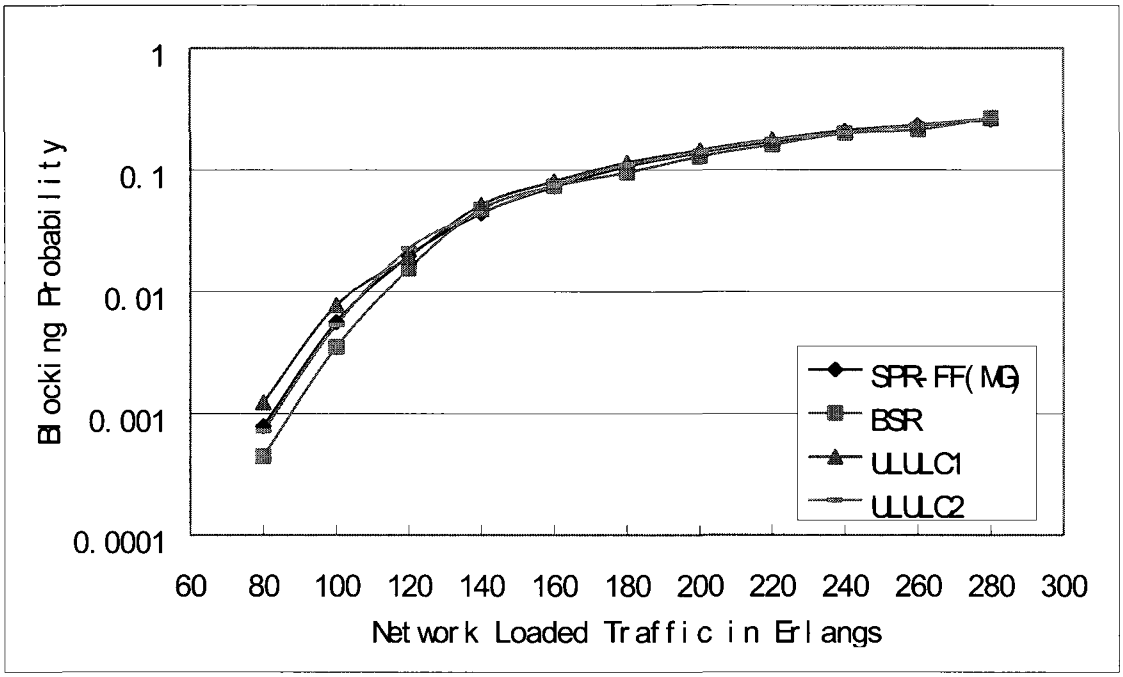

Figure 19: Blocking probabilities in the 14-node network

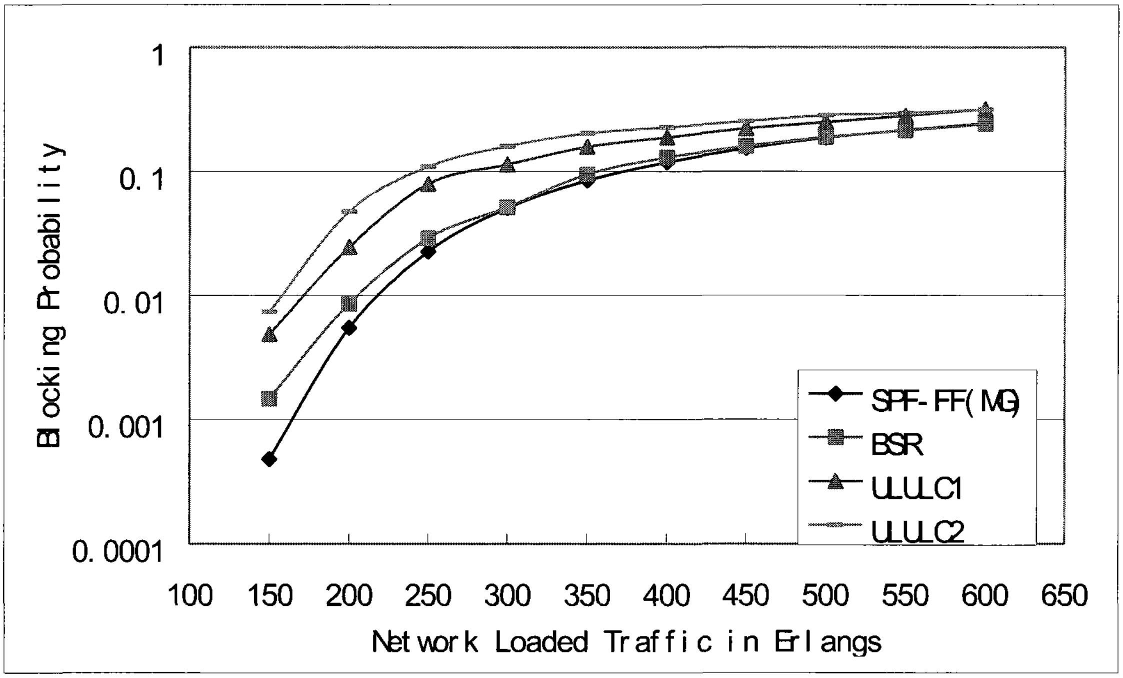

Figure 20: Blocking probabilities in the 28-node network

From the simulation results we can see that the ULULC1 &2 schemes do not perform as well as the BSR and SPR-FF (MG) schemes. The reason is that the hop count is a very important factor in dynamic RWA. The BSR scheme always takes the shortest paths, the link utilization is only used as a factor to select a path between several paths of equal length. When the Multiplier Guide is used to compute shortest paths, we see from all the given examples, that the paths obtained also have the lowest hop counts. However, the ULULC1 &2 schemes use link utilization as the only factor to determine paths, the hop count factor is not considered. Therefore, the performance of these two schemes are not good.

In the following, we mainly compare our SPR-FF (MG) scheme with the BSR scheme. In the 14-node NSFNET network, the BSR scheme slightly outperforms the SPR-FF (MG) scheme. However, in the 28-node Pan-European network, the BSR scheme performs worse than the SPR-FF (MG) scheme. In general the performances of the two schemes are close.

The BSR scheme has stronger assumptions than the SPR-FF (MG) scheme. First, the BSR scheme needs to obtain the link utilizations from the simulations. If this scheme is implemented in the real networks, link utilizations need to be measured in real time, which is hard. In the SPR-FF (MG) scheme, the Multiplier Guide is computed only once for a given average traffic matrix.

Second, in the BSR scheme, under different traffic loads, the link costs obtained by iterative simulations are different. Thus, if the traffic load changes, the paths have to be reconfigured. In the SPR-FF (MG) scheme, the Multiplier Guide can be applied to different traffic loads. As a result, even if the two schemes have similar performances, considering their different assumptions, the SPR-FF (MG) scheme is better.

5.5 Conclusions

We ran simulations and observed the performance of three two-stage dynamic RWA schemes in two different network topologies. We compared the results from these schemes using hop count and Multiplier Guide for paths planning. We found from simulations that, when the Multiplier Guide is used for paths planning in the SPR-FF, FAR-FF and LLR-FF schemes, the network performance is improved by having a lower blocking probability. In the two topologies we tested, the improvements are considerable when the dynamic demands obey the distribution of the traffic demand matrix. When the distribution of the dynamic traffic demands is different from the original traffic demand matrix, the improvement due to the Multiplier Guide is smaller but it still exists. We conclude that the Multiplier Guide can be used to improve the performance of the SPR-FF, FAR-FF and LLR-FF schemes.

We also compare performance of schemes that use the Multiplier Guide as link cost with schemes that use link utilization to decide link cost. It is found their performances are similar. However, the schemes that use the Multiplier Guide as link cost are easier to implement.

Chapter 6

Applying Multiplier Guide to a One-Stage Scheme

In Chapter 5 the Multiplier Guide was applied to several two-stage schemes and improvements of these schemes were found. In this chapter we apply Multiplier Guide to a one-stage scheme called Modified Dijkstra (M_Dijkstra). In this scheme the routing and the wavelength assignment sub-problems are solved jointly, and the entire solution space is searched. We expect that the Multiplier Guide can improve the performance of this one-stage scheme, since it can improve the performance of several two-stage schemes. We run simulations to find the result.

6.1 One-Stage Scheme

In Chapter 5 three two-stage schemes SPR-FF, FAR-FF and LLR-FF were studied. In these schemes, the RWA problem is solved by choosing a path from a set of fixed paths first and then choosing an available wavelength along the path. These schemes are constrained because only a limited number of paths are used. The performance of these schemes may not be optimal, since only a limited part of the entire solution space is searched.

In this chapter we want to study a one-stage RWA scheme that solves the routing and the wavelength assignment sub-problems jointly and considers all the possible paths. In such a scheme the traffic demand is always accommodated if a feasible RWA solution exists in the entire network. The solution is found by running a dynamic shortest path algorithm in the network each time a dynamic traffic demand comes. This scheme needs to know the status of the entire network and has very high computation cost, which makes it difficult to be implemented. This scheme often outperforms the two-stage schemes because it is not constrained.

The one-stage scheme has been studied previously in the literature. In [15] a one-stage scheme called Adaptive Unconstrained Routing (AUR) is proposed. In the AUR scheme the WDM network is transformed into several wavelength layers that contain all the solutions. Then the RWA problem is solved by finding the minimal cost path in all these layers. Several variations of the AUR scheme are proposed and compared. These variations are AUR/Spread, AUR/Random, AUR/Pack, AUR/Fixed and AUR/Exhaustive. The first four variations search a solution in different wavelength layers in a certain sequence and terminate once a solution is found in one layer. These variations have lower computation cost but they may not find the minimal cost path. The AUR/Exhaustive variation searches all the wavelengths and can finally find the path with the minimal cost. The AUR/Exhaustive scheme is the scheme we want to implement. However, the computation cost of this scheme is very high. In [16], a layered graph that is a revision of the wavelength layers in AUR scheme is proposed. Also in [16], a Modified Dijkstra algorithm is proposed, which can be used to solve the minimal path cost problem in the layered graph. This algorithm has the same function as the AUR/Exhaustive scheme and it has lower computation cost.

In this chapter we implement the one-stage scheme that uses a layered graph to represent the WDM network and uses the Modified Dijkstra algorithm to solve the minimal cost path problem. We call it the Modified Dijjkstra (M_Dijkstra) scheme. We compare the performance difference between using hop count and Multiplier Guide as link cost in this scheme.

6.2 Layered Graph

The generation of the layered graph is presented in the following. A WDM network without wavelength conversion capability can be seen as several independent networks, one for each wavelength layer. The layered graph is based on the same idea. To generate the layered graph, at first the original network is duplicated into n separate wavelength layers, where n is the total number of wavelength the network has. In each wavelength layer, the network topology is identical to that in the original network. Initially, these wavelength layers are not connected. When a path has to be found between two nodes in the original network, the two nodes are added to the wavelength layers by connecting them to their duplication nodes in each wavelength layer. When the two additional nodes are added to the wavelength layers, the graph becomes the layered graph. An example of the layered graph generation is shown as follows. Figure 21 shows the original network with 3 available wavelengths. If a path has to be found between nodes A and D, the corresponding layered graph is shown in Figure 22.

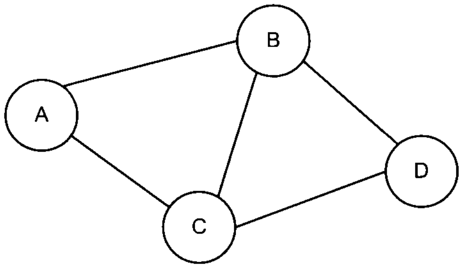

Figure 21: The original network

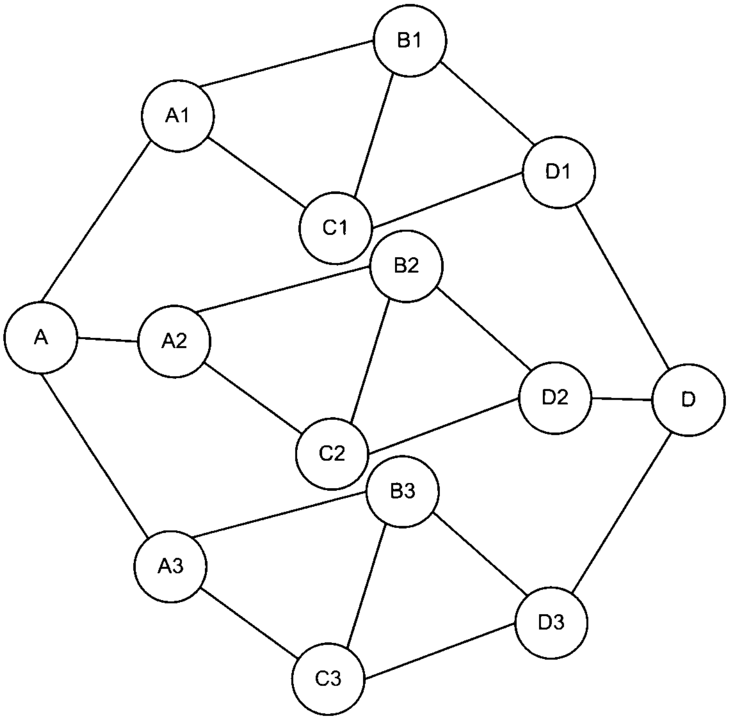

Figure 22: The layered graph

In Figure 22, nodes Ai, Bi, Ci, Di and the links interconnecting them comprise the ith wavelength layer. Nodes A and B are the two additional nodes that represent the source and destination nodes of the traffic demand. In the layered graph, each path between Nodes A and B is an RWA solution for this node pair. The layered graph contains all the RWA solutions.

Link costs in the layered graph are set as follows: each wavelength layer is a duplication of the original network, so the links in each wavelength layer have the same costs as the corresponding links in the original network. The links connecting the two additional nodes and their duplication nodes in the wavelength layers have zero cost.

6.3 Modified Dijkstra Algorithm

The Modified Dijkstra Algorithm [16,31] is a modification of the Dijkstra shortest path algorithm. This algorithm makes use of the particular structure of the layered graph and reduces its computational complexity.

To define the Modified Algorithm, the notations are first presented. Let N and W denote the total number of nodes in the original network and the total number of wavelengths, respectively. Thus, in the layered graph, there are $WN+2$ nodes, including the two additional nodes. The source and destination nodes are labeled as 0 and $WN+1$. The nodes in the k th layer are labeled as $(k-1)N+1,(k-1)N+2,\cdots,kN$. The cost of a directed link in the layered graph is denoted by c_{ij}, where i is the transmitting node and j is the receiving node. The shortest path from node 0 (source node) to node i is denoted by p_i and its cost is denoted by D_i. The permanent set is denoted by S.

The Modified Dijkstra algorithm is defined in the following. The original algorithm contains an error [31]; its steps 3 and 4 should be reversed. We present here the corrected algorithm.

Step 1 (Initialization):

1) $S := \{0\}$;

2) $D_i = c_{0i}$, $\forall i \in \Omega - S$; if $D_i < \infty$, then $p_i = \{0\} \cup \{i\}$ else $p_i = \Phi$;

3) $R_k := \min\{D_i : i \in \{(k-1)N+1,\cdots,kN\}\}$, I_k is the node number corresponding to R_k;

Step 2 (Designation of a New Permanent Label):

1) $R_w := \min\{R_k, k \in \{1,2,\cdots,W\}\}$, I_w is the node number corresponding to R_w.

2) If $D_{Iw} < \infty$, $S := S \cup \{I_w\}$; else return "cannot find a shortest path with finite cost," STOP;

3) If $I_w = NW+1$, return "p_{Iw} is the solution," STOP.

Step 3 (Update D_i and p_i, node i is on layer w):

If $i \in \{(w-1)N+1,\cdots,wN\}$, $i \notin S$ and $D_{lw}+c_{lwi} < D_i$, then $D_i := D_{lw}+c_{lwi}$, $p_i := p_{lw} \cup \{i\}$.

Step 4 (Update R_w):

If $\{i : i \in \{(w-1)N+1,\cdots,wN\}, i \notin S\} = \Phi$,

then $R_w := \infty$, else $R_w := \min\{D_i : i \in \{(w-1)N,\cdots,wN\}, i \notin S\}$. Go to Step 2.

Because the duplication links in each layer have the same weight, several minimal cost paths having the same total cost can often be found. Here the ties are broken randomly.

The complexity of the Modified Dijkstra algorithm is $O(\max\{N,W\}NW)$. It is lower than the complexity of the Dijkstra algorithm for the layered graph, which is $O((NW)^2)$.

6.4 Simulation Results

We have run simulations in the 14-node NSFNET network and the 28-node Pan-European network. In each network we generate non-uniform arrivals according to the traffic demand matrix and use the Multiplier Guides obtained from the static RWA problem. In the 14-node NSFNET network, only the first traffic matrix and the corresponding Multiplier Guides are used.

The simulation results are shown in the followings. In Figure 23, the blocking probabilities versus offered traffic loads for the M_Dijkstra and LLR-FF schemes in the 14-node NSFNET network are shown. For both schemes, the Multiplier Guide and Hop count are used for path computation. The results for the same schemes in case of the 28-node Pan-European network are shown in Figure 24.

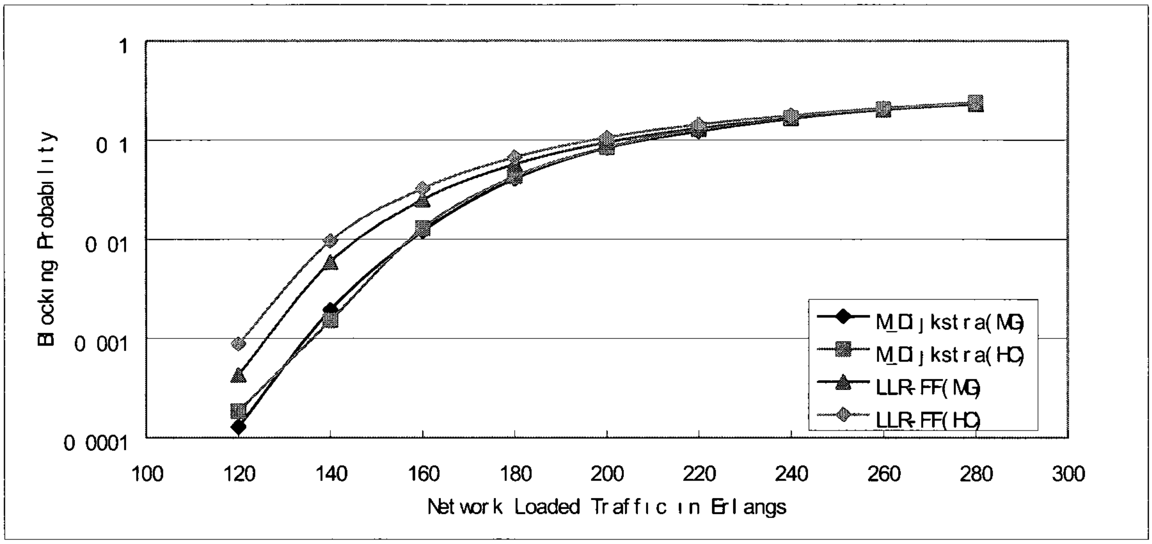

Figure 23: Blocking probability in the 14-node network

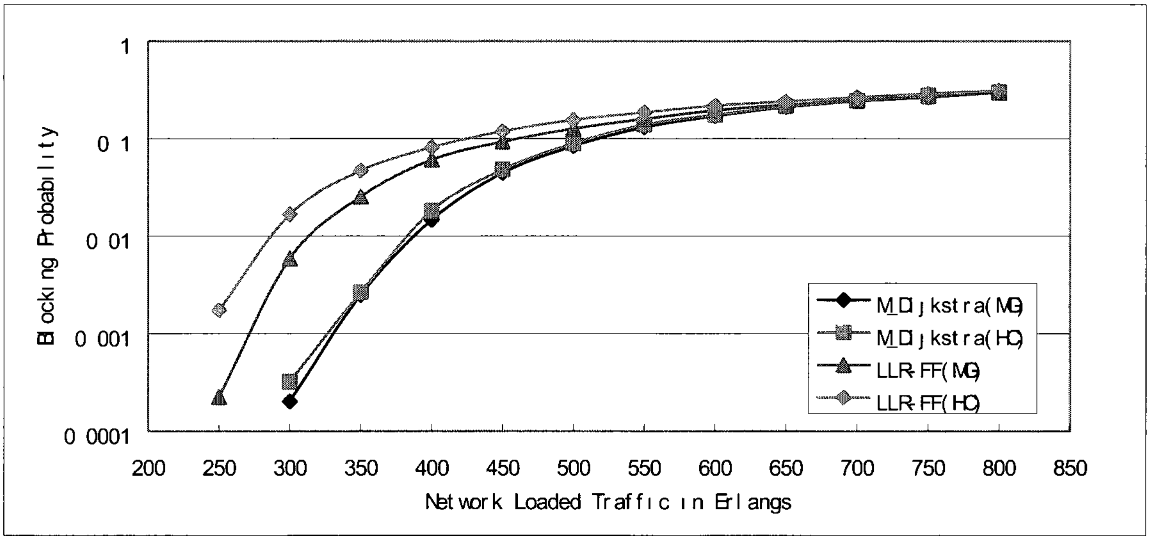

Figure 24: Blocking probability in the 28-node network

Based on these simulation results, several observations can be made. We know that among the three two-stage schemes studied in Chapter 5, the LLR-FF scheme performs the best. Here it can be seen that the one-stage scheme M_Dijkstra outperforms the LLR-FF scheme. This is because the M_Dijkstra algorithm searches the entire RWA space while the LLR-FF scheme does not.

Another important observation is that when the Multiplier Guide is applied to the M_Dijkstra scheme, the performance is not improved. From the simulation results it can be seen that the M_Dijkstra scheme using Multiplier Guide (MG) and Hop count (HC) have similar performance. Only when the arrival rate becomes low, the M_Dijkstra(MG) has a slightly better performance. Considering that the calculation of the Multiplier Guide in the static case is not easy and the performance improvement is very small, it is not economical to apply the Multiplier Guide to the M_Dijkstra scheme.

6.5 Conclusions

In this chapter we implemented a one-stage scheme called Modified Dijkstra that converts the WDM network into a layered graph and uses the Modified Dijkstra algorithm to find the minimal cost path. Then we run simulations to see the performance difference between using Multiplier Guide and Hop Count as link cost in this scheme. The result is that using Multiplier Guide as link cost does not improve the performance of the M_Dijkstra scheme.

Chapter 7

Applying Multiplier Guide to K-Shortest Paths Routing First-Fit scheme

In this chapter we propose a new scheme K-Shortest Paths First-Fit (KSP-FF) and apply the Multiplier Guide to it. In this scheme we use a set of fixed paths for routing. We implement a k-shortest paths algorithm to find the first k shortest paths in the network and use them to accommodate the dynamic demands. We run simulations to find out whether the Multiplier Guide can improve the performance of this scheme. With the k-shortest paths algorithm, we are able to find all the equal cost shortest paths for each node pair. Thus, we can propose another new scheme Equal Cost Shortest Paths First-Fit (ECSP-FF) and test it with the Multiplier Guide.

7.1 K-Shortest Paths First-Fit Scheme

In Chapter 5 we studied and implemented the FAR-FF and LLR-FF schemes. In these schemes, two link disjoint paths are constructed to accommodate the traffic for each node pair. Now we want to construct a set of k-shortest paths to accommodate the traffic for each node pair. The k-shortest paths means a total of k paths that are ranked from the first shortest to the k-th shortest. The k-shortest paths are shorter than any other paths. The k-shortest paths are allowed to share links.

We consider using k-shortest paths for the following reasons. One reason is that by using link-disjoint paths, it is possible that the second path is too long, since it is disjoint from the first path. This may lead to bad performance because the long path consumes too many network resources. Another reason is that here we want to construct more than two paths for every node pair. If the paths are required to be link-disjoint, much less paths can be found.

In the K-Shortest Paths First-Fit scheme we propose, a set of the k-shortest paths are constructed for each node pair. When a demand comes, the k paths are searched from the shortest to the longest. The shortest of the k paths that has an available wavelength is used. First-Fit is used as tie-breaker when several available wavelengths exist on a path.

7.2 Finding K-Shortest Paths in a Graph

We present the k-shortest paths algorithm here without rigorous mathematical proofs. The details of this algorithm can be found in [17] and [18]. We first explain two concepts: the Tree of Paths and the New Path Generation Method. Then, the k-shortest paths algorithm is presented.

7.2.1 The Tree of Paths

Many different paths having the same source and destination nodes are found during the process of finding the k-shortest paths. A Tree of Paths is needed to store all these paths. The Tree of Paths is used in the new paths generation process to prevent getting new paths that actually have already been generated, as presented in Section 7.1.2. The construction of the Tree of Paths is presented here first.

The Tree of Paths consists of nodes. Each node has a number indicating the corresponding node in the network. A node may have one or many son nodes but can only have one father node. We use several nodes connected in a line to represent a path. It is stipulated that, in the line of nodes, the source node of the path is the node that has no father node while the destination node of the path is the last node that has no son node.

The Tree of Paths is initialized to having a root node only, which is the source node that all the paths share. Each time a new path is generated, it is added to the tree. Each path from the root to a leaf on the Tree of Paths represents a path in the network. We use *T* and *P* as two pointers indicating two nodes, one on the Tree of Paths, and the other on the path to be added to the tree, respectively. Thus, these two nodes are denoted by nodes *T* and *P*. When a new path is to be added to the Tree of Paths, the following algorithm is applied:

Algorithm for Adding a Path to the Tree of Paths (Algorithm APTP):

```
1. Assign T to point to the root node of the Tree of Paths. Assign P to point to the source node of the path. (Here the two nodes have the same index)
2. While (node P has a son node)
    Assigned P to point to node P's son node
    If (node T has a son node that has the same index as node P)
        Assign T to point to node T's son node that has the same index as node P
    Else
        Create a new node that has the same index as node P
        Make this new node the son node of node T
        Assign T to point to node T's new son node
    End
End
```

To give a more intuitive illustration of this algorithm, an example is shown. Figure 25 shows a network. Three different paths connecting nodes 1 and 6 can be found. They are path 1: <1,2,3,6>, path 2: <1,2,3,4,6> and path 3: <1,2,5,3,6>. Figure 26 shows three Trees of Paths from left to right. The first tree contains only path 1. The second tree contains both paths 1 and 2. The third tree contains paths 1, 2 and 3.

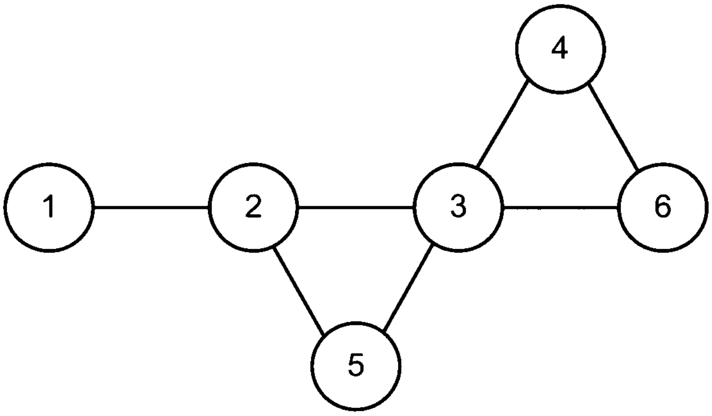

Figure 25: An example network

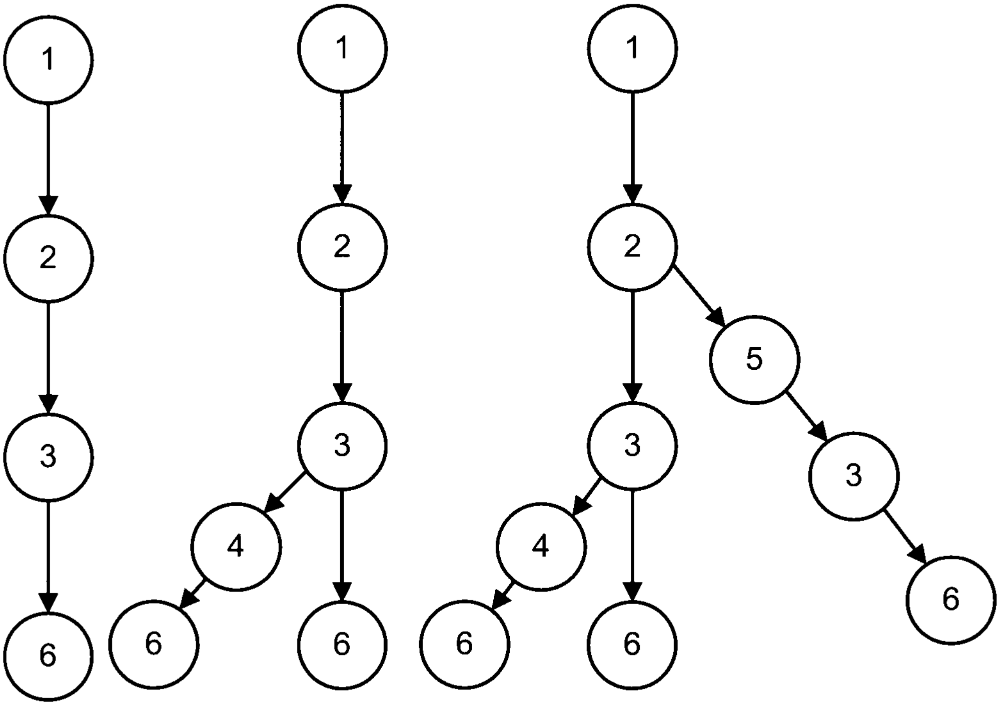

Figure 26: Three Trees of Paths

7.2.2 The New Path Generation Method

In order to find the k-shortest paths, we need to generate new paths from existing paths. A new path is generated from an existing path as follows: The new path and the existing path have the same source and destination nodes. The new path traverses the same nodes and links as the existing path from the source node until a so-called deviation node. On the deviation node, the new path traverses a different link from the existing path to a different node. Then the new path continues from the different node to the destination node. The new path consists of two parts: the path from its source node to the deviation node, which is called head path; and the path from the deviation node to its destination node, which is called tail path. The deviation node of a new path has to be recorded, which is useful in the k-shortest paths algorithm described in Section 7.1.3.

Consider the network example shown in Figure 25 and the existing path 1: <1,2,3,6>. Path 2: <1,2,3,4,6> is a new path generated from path 1 with node 3 as deviation node. Path 3: <1,2,5,3,6> is a new path generated from path 1 with node 2 as deviation node.

Once the existing path and a deviation node are chosen, the new path is generated as follows: The head path of the new path is the part of the existing path going from its source node to its deviation node. The tail path of the new path is found by removing the outgoing link of the deviation node that the father path traverses and then running the Dijkstra algorithm to find a path between the deviation node and the destination node. Finally the head and tail paths are combined to get the new path.

We want to use an existing path to generate a new path. However, if the existing path is generated from another path, by using the scheme above, we may generate a new path that is actually the path from which the existing path was generated. An example of this situation is the following. The existing path is <1,2,4,3,6>, which is generated from the path <1,2,3,6>. If node 2 is chosen to be the deviation node on path <1,2,4,3,6>, we want to get the new path {1,2,5,3,6}, which has not been generated before. However, by using the scheme above, we remove link <2,4> and then run the shortest path algorithm to get a path between nodes 2 and 6. As a result we get path <1,2,3,6>.

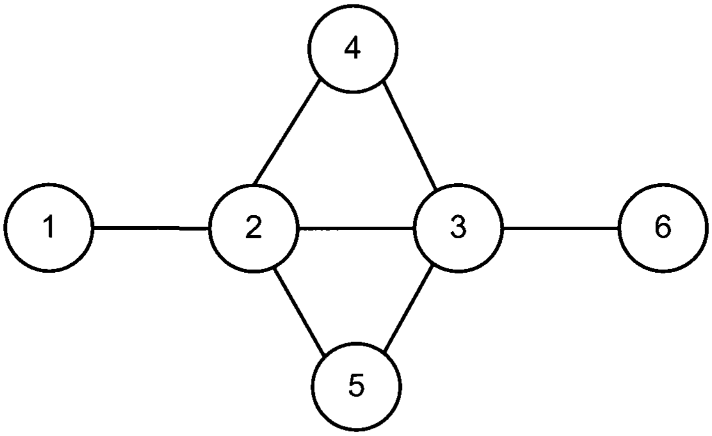

Figure 27: An example network

We can use the Tree of Paths to prevent generating a path that has already been generated. First we add all the paths we generated to the Tree of Paths by using the algorithm presented in Section 6.2.1. Once we have chosen an existing path and a deviation node on it to generate a new path, the New Path Generation Method processes as follows.

First we find the node on the Tree of Paths that represents the deviation node of the existing path. This node can be found by starting with the root node of the Tree of Paths and then traversing a path, which is the head path of the new path, on the Tree of Paths, until reaching the deviation node. Once the deviation node on the Tree of Paths is found, we check its son nodes. Then we remove the links in the network that have the deviation node as the transmitting node and the deviation node's son nodes as the receiving node. After these links are removed, we run the Dijkstra algorithm to find a shortest path between the deviation node and the destination node, which is the tail path. Finally the head path and tail path are combined to form the new path.

Consider the example network in Figure 27. After the paths <1,2,3,6> and <1,2,4,3,6> are generated, the Tree of Paths containing them becomes the tree shown in Figure 28. Now we choose path <1,2,4,3,6> and node 2 the deviation node to generate a new path. Using the method described above, we find that the dashed node 2 is the deviation node on the Tree of

Paths, which is shown in Figure 28. Then links <2,4> and <2,3> are removed from the network. Finally the desired new path <1,2,5,3,6> is found.

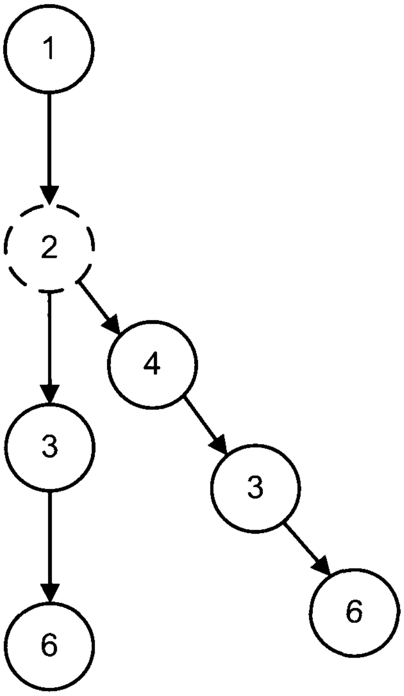

Figure 28: An example of Tree of Paths

It can be proven that by using the New Path Generation Method, the new path generated is always no shorter than the paths that have already been generated previously. This is an important lemma in finding the k-shortest paths.

7.2.3 K-Shortest Paths Algorithm

After the notions of Tree of Paths and New Path Generation Method are introduced, we can now present the algorithm of finding the k-shortest paths. The general idea of this algorithm is that we use a permanent set to store the k-shortest paths and a temporary set to store candidate paths. The candidate paths are the paths that have been generated but have not entered the permanent set. First we find the shortest path between the source and destination nodes. We make the source node of this path the deviation node and put it into the temporary set. Then we execute a loop for k times: In each repetition we move the shortest path in the temporary set to the permanent set. Then we use this path to generate several new paths and add them to the temporary set. After the loop is done k times, the permanent set contains k paths and they are the k-shortest paths.

The way to generate new paths by using the shortest path *p* removed from the temporary set is described in the following. We generate several new paths by using path *p* and choosing the deviation nodes to be the nodes contained on *p*'s tail path. Those nodes include path *p*'s deviation node and intermediate nodes but exclude path *p*'s destination node. Once a deviation node of path *p* is chosen, the New Path Generation Method presented in Section 7.2.2 is used to generate the new path. We do not use the nodes contained in path *p*'s head path (excluding the deviation node) as deviation nodes to generate new paths. This is because these paths have already been generated previously.

The K-Shortest Paths algorithm is described in the following. While running the algorithm, we use a Tree of Paths to store all the paths generated. We use *P* and *T* to denote the permanent set and the temporary set respectively. We use a pointer *t* to indicate a node on a path.

K-Shortest Paths Algorithm

```
1. Find the shortest path between the source and destination nodes. Make the source node
the path's deviation node
2. Add this path to the Tree of Paths and set $T$
3. Loop k times
    Move the shortest path $p$ in set $T$ to set $P$ (if set $T$ is empty, then terminate)
    Assign $t$ to point to the deviation node of path $p$
    While (node $t$ is not the destination node of the path)
        Use the New Path Generation Method to generate a new path of path $p$ with node
```

```
        $t$ the deviation node
        If the new path can be found
            Use Algorithm APTP to add the new path to the Tree of Paths
            Add the new path to set $T$
        End
        Assign node $t$ to be its son node on path $p$
    End
End
```

7.3 Simulation Results

We set k to different values and observe the performance of the KSP-FF scheme. First we compare the performances for different values of k from 2 to 16. Then we set k to 2 and compare this scheme with the FAR-FF scheme. The two schemes both use two paths for each node pair, but the paths are constructed differently. Then we set k to be large enough that the set of k-shortest paths contains all the possible paths in the network. Thus, the KSP-FF scheme is equivalent to the one-stage M_Dijkstra scheme. In this case, we compare the KSP-FF scheme with the M_Dijkstra scheme. Finally, we introduce a new scheme called Equal Cost Shortest Paths First-Fit that uses all the equal cost shortest paths to accommodate the traffic.

7.3.1 The Trend of Increasing k in KSP-FF Scheme

In both the 14-node NSFNET network and the 28-node Pan-European network, we run simulations of the KSP-FF scheme with k being 2, 3, 4, 8 and 16. In each case we use Multiplier Guide and hop count for path constructions. The simulation results from the 14-node NSFNET network and the 28-node Pan-European network are shown in Figures 29 and 30, respectively.

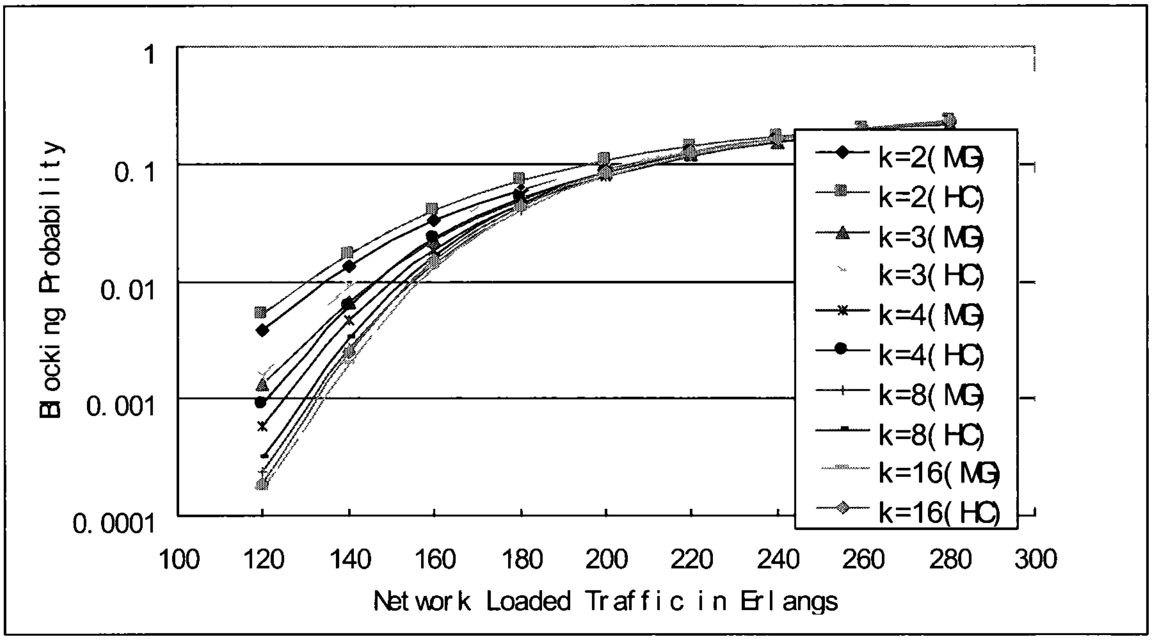

Figure 29: The Results of the KSP-FF scheme with different values of k in the 14-node network

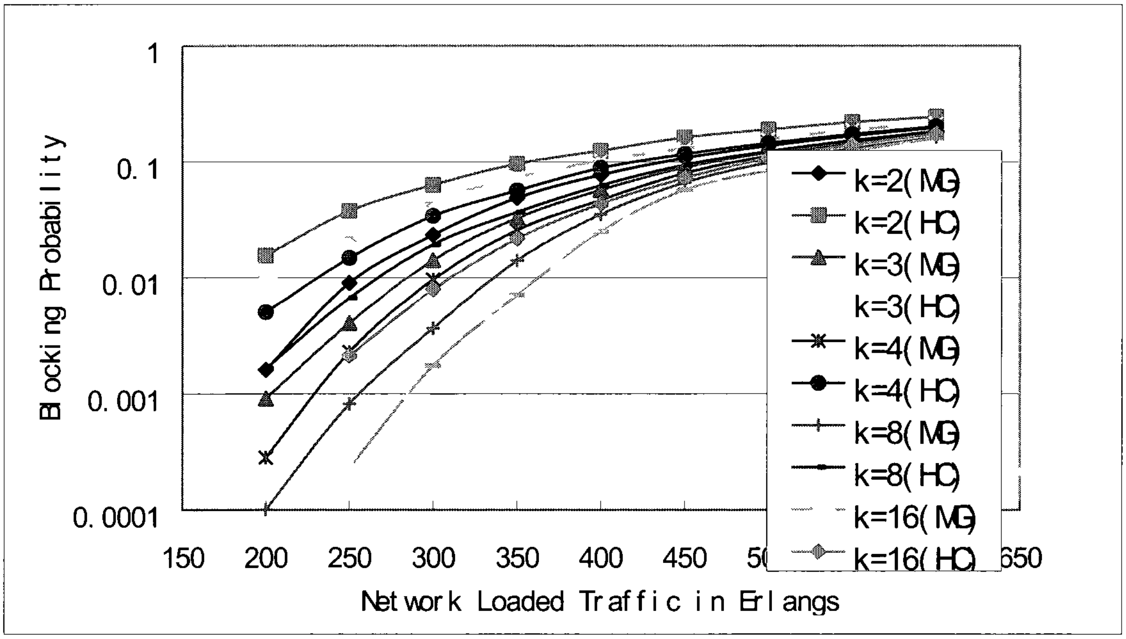

Figure 30: The Results of the KSP-FF scheme with different values of k in the 28-node network

Several observations can be made about Figures 29 and 30. First, as k in the KSP-FF scheme increases, the performance under medium and low arrival rates becomes better. Second, when k becomes large, the performance of this scheme does not improve much as we increase k. From this we can expect that the performance of this scheme will reach a certain limit as k becomes large enough. Third, some performance improvement can be seen when the Multiplier Guide is applied to the KSP-FF scheme with all values of k. However, when k becomes large, the improvement becomes smaller.

7.3.2 Comparing KSP-FF (k=2) and FAR-FF Schemes

When the KSP-FF (k=2) scheme is compared with the FAR-FF scheme, as shown in Figures 31 and 32, the following facts can be observed. We mainly focus on the two schemes with the Multiplier Guide applied. In the 14-node NSFNET network, the KSP-FF (k=2) scheme outperforms the FAR-FF scheme when the arrival rate is no lower than 180. When the arrival rate becomes lower, the FAR-FF scheme performs better. In the 28-node Pan-European network, the KSP-FF (k=2) scheme outperforms the FAR-FF scheme when the arrival rate is no lower than 500. At lower arrival rates the FAR-FF scheme performs better. This is because when the arrival rate is high, using 2-shortest paths is better. Since the longer path of the two link-disjoint paths may be too long and costly. When the arrival rate is low, using two link-disjoint paths is better. Since the 2-shortest paths may share some links that may block both paths. Considering the fact that the real networks are operating under relatively low arrival rate (the blocking probability is at around 0.01), we consider that the FAR-FF scheme is better than the KSP-FF (k=2) scheme.

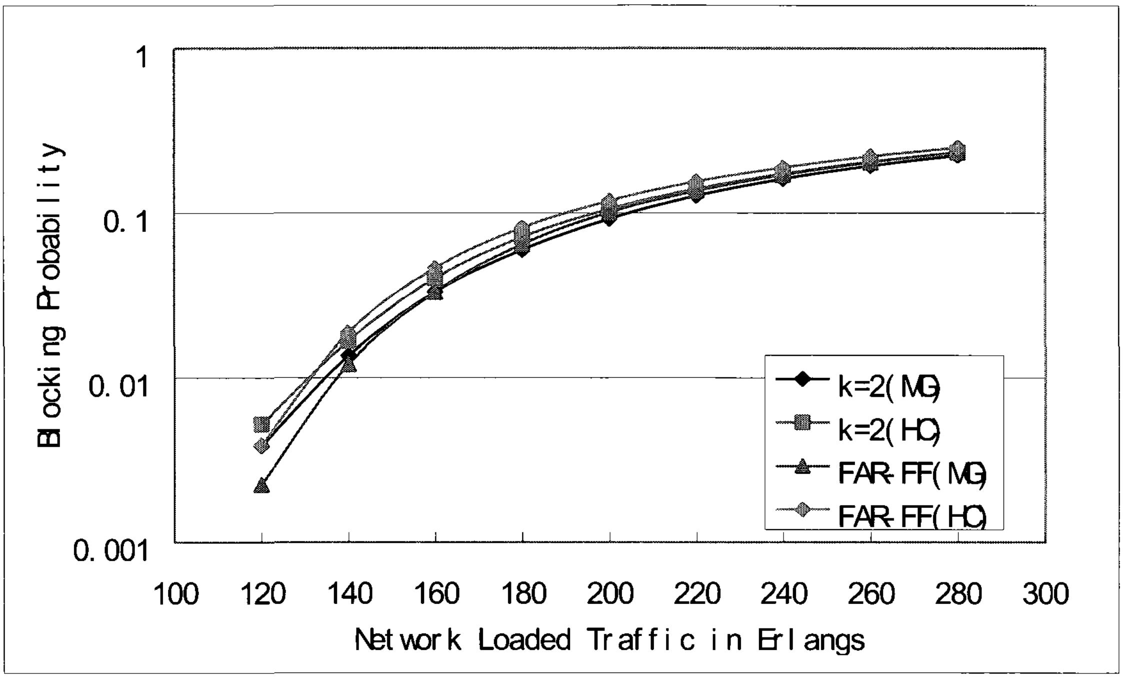

Figure 31: Comparison of KSPF-FF (k=2) and FAR-FF schemes in the 14-node network

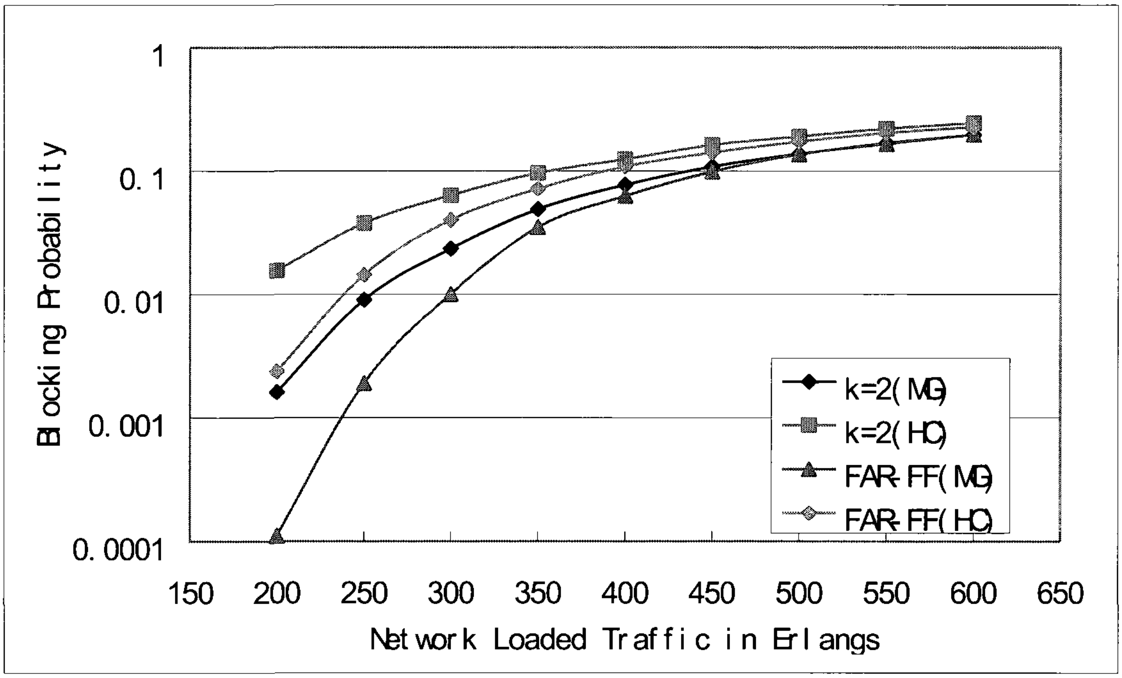

Figure 32: Comparison of KSPF-FF (k=2) and FAR-FF schemes in the 28-node network

7.3.3 Comparing the KSP-FF Scheme with Sufficient Large k with the M_Dijkstra Scheme

Here we want to set k to be large enough that all the possible paths are considered. Then we compare the performance of the KSP-FF scheme with the one-stage M_Dijkstra scheme. In the 14-node NSFNET network, it can be calculated that between each node pair there are at least 40 and at most 120 paths. Thus we set k to be 120 and run simulations in this network. In the 28-node Pan-European network, we find that there are at least 10000 different paths between each node pair. The number of paths to be constructed in this network is too large. Thus we do not use this network to run simulations.

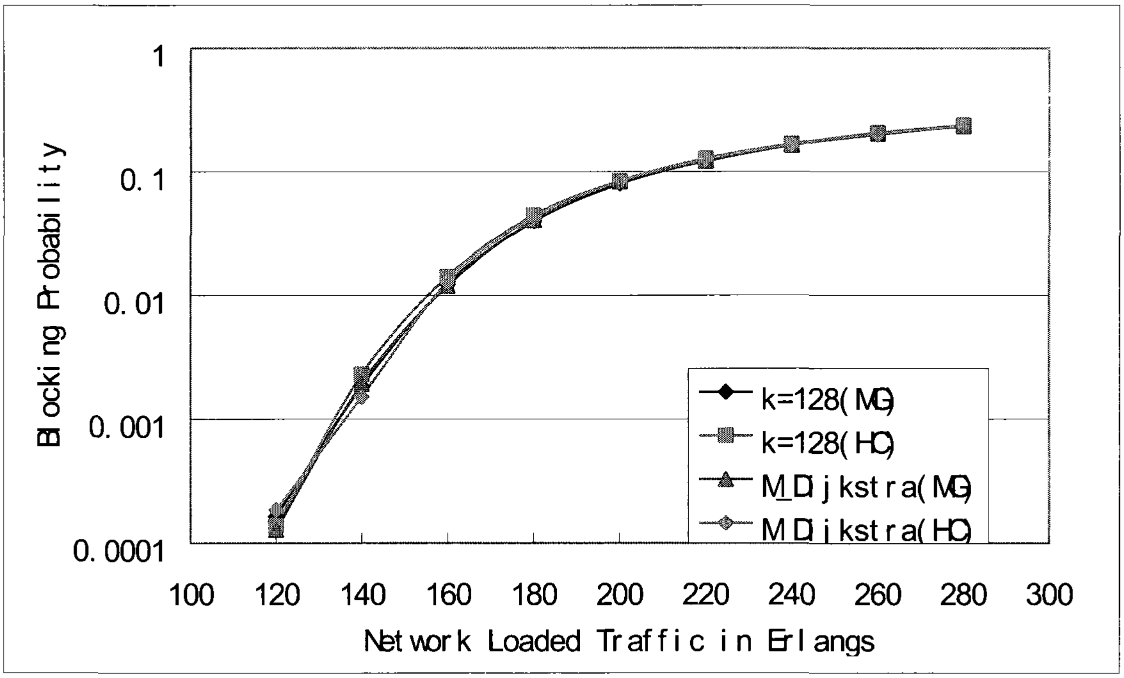

Figure 33: Comparison of KSP-FF (k = 120) and M_Dijkstra schemes in the 14-node network

From Figure 33 we see that the KSP-FF (k=120) and M_Dijkstra schemes have very similar performance. And when the Multiplier Guide is applied to the KSP-FF (k=120) scheme, the performance is not improved. The same result happens when the Multiplier

Guide is applied to the M_Dijkstra scheme. This finding verifies the results obtained in Chapter 6, since in this network the KSP-FF (k=120) and M_Dijkstra schemes are actually the same.

7.3.4 Equal Cost Shortest Paths First-Fit (ECSP-FF) Scheme

With the k-shortest paths algorithm, for any two given nodes in the network, we can find the k shortest paths connecting them. Thus, we are able to find all the equal cost shortest paths. Here we propose the Equal Cost Shortest Paths First-Fit scheme. In this scheme, all the equal cost shortest paths between a node pair are used to accommodate the traffic. When the hop count is used for path computation, the shortest paths having the same number of hops are used. When the Multiplier Guide is used for path computation, since the multiplier values are real numbers, the paths having costs no greater than 1.1 times the cost of the shortest path are used. Since these paths are equal, the least congested one among them is chosen to accommodate the traffic. The First-Fit scheme is used to determine which wavelength to use.

The numbers of equal cost shortest paths found in the two topologies are shown as the following. In the 14-node NSFNET network, when the hop count is used, on average 1.28 equal cost shortest paths between a node pair can be found. When the Multiplier Guide is used, the average number is 1.24. In both cases, the maximum number of equal cost shortest paths found for a given node pair is 3 while the minimal number is 1. In the 28-node Pan-European network, by using the hop count, on average 2.72 equal cost shortest paths can be found. The maximum number of equal cost shortest paths is 26 while the minimal number of equal cost shortest path is 1. By using the Multiplier Guide, in average 3.08, maximum 3.08 and minimum of 1 paths are found.

Figure 34 shows the results for the 14-node NSFNET network. Figure 35 shows the results for the 28-node Pan-European network. The results for the SPR-FF and LLR-FF schemes are also shown for comparison.

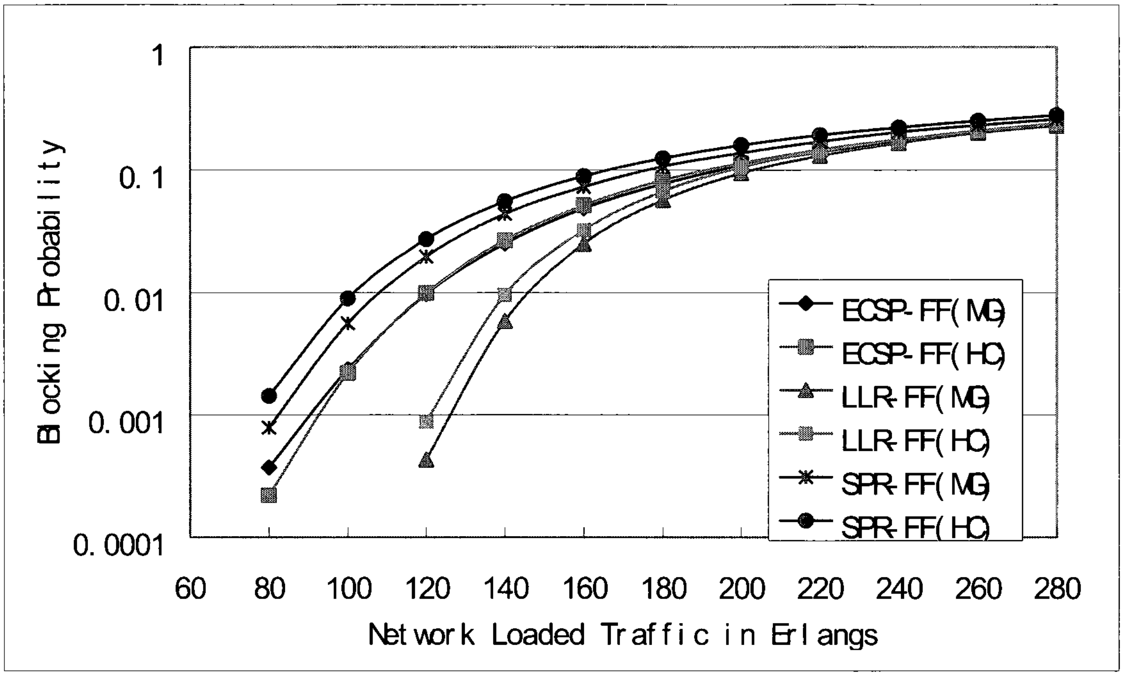

Figure 34: Results of the ECSP-FF scheme from the 14-node network

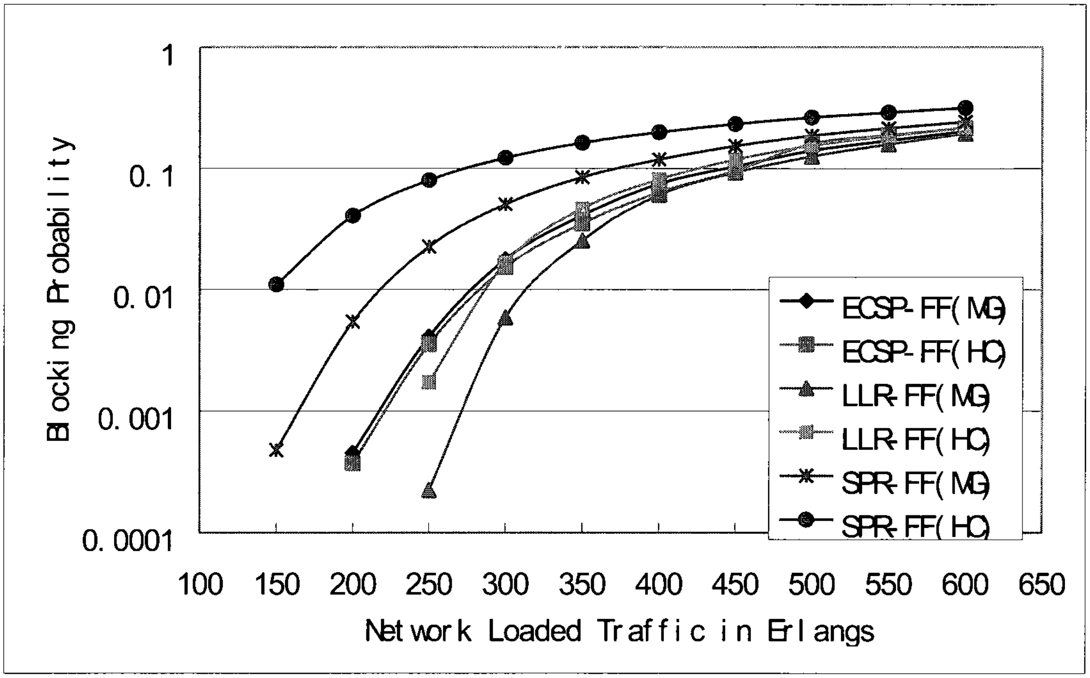

Figure 35: Results of the ECSP-FF scheme from the 28-node network

It can be seen that when the Multiplier Guide is used in the ECSP-FF scheme, the performance improvement is not noticeable. The ECSP-FF (MG) and ECSP-FF (HC) schemes always have similar performance. The possible reason for this is that the main function of the Multiplier Guide is to determine the best path among several equal cost ones. In the ECSP-FF scheme all the equal cost paths are used, the difference between using the Multiplier Guide and hop count becomes smaller.

In both networks, the ECSP-FF scheme outperforms the SPR-FF scheme, but it performs worse than the LLR-FF scheme. Note that in the 14-node network the ECSP-FF (MG) and ECSP-FF (HC) schemes use in average 1.24 and 1.28 paths, respectively. However, in the 28-node network, the two numbers are 3.08 and 2.72. We can infer that in large network topologies, the ECSP-FF scheme may lead to a large number of paths in use but the performance is not proportionally improved. The possible reason for this is that when the network is big, between some particular node pairs, a large number of equal cost shortest paths may be found. All these paths are used in the ECSP-FF scheme, which may raise the average number of paths in use. However, between these particular node pairs, there may not be lots of traffic. If so, these paths are barely used and it is a waste. Another reason is that these equal cost shortest paths may not be link-disjoint. It has been shown previously that using link-disjoint paths is better. The LLR-FF scheme is better than the ECSP-FF scheme, even though in some topologies the former scheme uses fewer paths.

7.4 Conclusions

In this chapter we have shown the k-shortest paths algorithm and the KSP-FF scheme. Our simulations have shown that the Multiplier Guide can improve the performance of the KSP-FF scheme. The performance of the schemes using 2-shortest paths and two link-disjoint paths are studied and it is found that the latter is better in most cases. The results generated from the KSP-FF scheme with sufficiently large k are compared with the results from the M_Dijkstra scheme and their performance is the same. We have also investigated the performance of the ECSP-FF scheme, which uses all the equal cost shortest paths. We have found that the Multiplier Guide cannot improve the performance of this scheme. And this scheme may not be suitable for large networks.

Chapter 8

Applying Multiplier Guide to RWA Schemes Including Wavelength Conversion

In Chapters 5, 6 and 7 we studied dynamic RWA problems in WDM networks without consideration of wavelength conversion. In this chapter we assume that the WDM networks have wavelength conversion capabilities. We apply the Multiplier Guide to several RWA schemes that include wavelength conversion. These RWA schemes include a one-stage scheme, called Shortest Path Algorithm in Wavelength Graph (SPAWG), and two schemes using fixed routes, called Shortest Path Routing with Wavelength Conversion (SPR-WC) and Disjoint Paths Routing with Wavelength Conversion (DPR-WC). We run simulations to see whether the Multiplier Guide can improve the performance of these schemes.

8.1 Wavelength Conversion Assumption

In previous chapters, we mentioned that wavelength conversion devices can break the wavelength continuity constraint in WDM networks and improve network performance. However, wavelength converters are is still very expensive. For this reason, the WDM networks are often equipped with a limited number of wavelength converters.

The literature mentions several ways to equip WDM network with a limited number of wavelength converters. In [19] and [4], it is assumed that some nodes in the WDM network have sufficient number of wavelength converters while other nodes have no wavelength converters. In [20] and [1], it is assumed that all nodes in the network have the same number of converters, but their number is insufficient.

In this chapter we take the second assumption. We assume that all nodes in the network have the same number of wavelength converters. In this case, when the number of converters installed at each node reaches to a certain value, which is often not very big, the network achieves a performance very similar to the case of full wavelength conversion [20]. In the following, we set the number of converters to this value.

8.2 Minimal Cost Semi-Lightpath Problem

Previously we introduced the term semi-lightpath for the case that a connection uses more than one wavelength along its path. If a connection is a semi-lightpath, it uses at least one wavelength converter along its path. Each wavelength converter is associated with a wavelength conversion cost. If a connection uses a wavelength converter in one of its intermediate node, the wavelength conversion cost is added to its total cost.

When a traffic demand occurs in a WDM network equipped with wavelength converters, a lightpath or semi-lightpath having the minimal cost has to be found to accommodate the demand. This is called the minimal cost semi-lightpath problem. To solve the problem, a scheme was proposed in [21]. In this scheme, the network is first transformed to a wavelength graph. Then an algorithm called Shortest Path Algorithm in Wavelength Graph is applied to find the minimal cost path.

8.2.1 Wavelength Graph

The generation of the wavelength graph is presented in the following. Similar to the generation of the layered graph described in section 6.2, at first the network is duplicated into *k* layers, where *k* is the total number of wavelength in the network. Then, if a node has wavelength converters, its duplicated nodes in all the layers are connected to each other. The links connecting these duplicated nodes represent wavelength conversion. By using these links the semi-lightpath can go from one wavelength to another. Note that the wavelength conversion links connecting the duplicated nodes share the same capacity of the converters. Finally, an additional layer of the network is added to the graph; this layer represents the nodes in the original network. Different from the other layers, in this layer the nodes are not connected to each other by links that exist in the original network. Each node in this layer is connected to all corresponding duplicated nodes in the other layers with links having zero cost. With the wavelength graph completed, a path connecting two nodes in the additional layer represents an RWA solution possibly including wavelength conversion.

An example is shown in the followings. Figure 36 shows the original network, its node C has wavelength converters installed. In the network there are 3 wavelengths on each link. Figure 37 shows an uncompleted wavelength graph where the network is duplicated into 3 layers and nodes C1, C2 and C3 are interconnected by links, which represent wavelength conversion. Figure 38 shows the final wavelength graph. Here the nodes A, B, C and D, representing the original model of the network, are added to the graph and the four nodes are connected to their duplicated nodes in every layer.

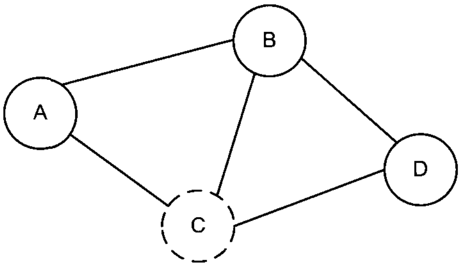

Figure 36: The original network

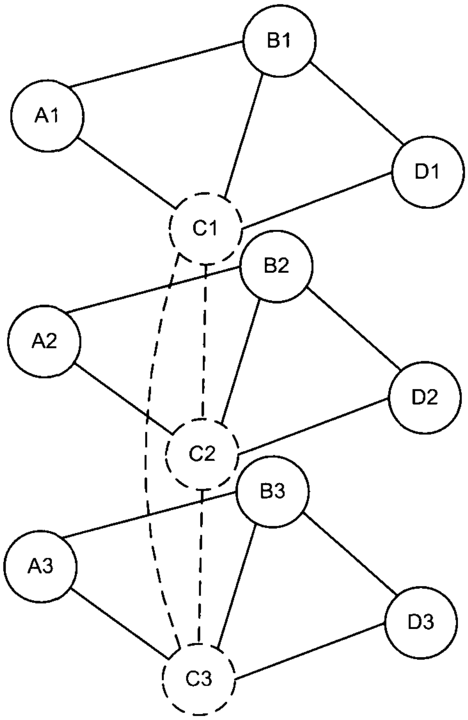

Figure 37: Uncompleted wavelength Graph

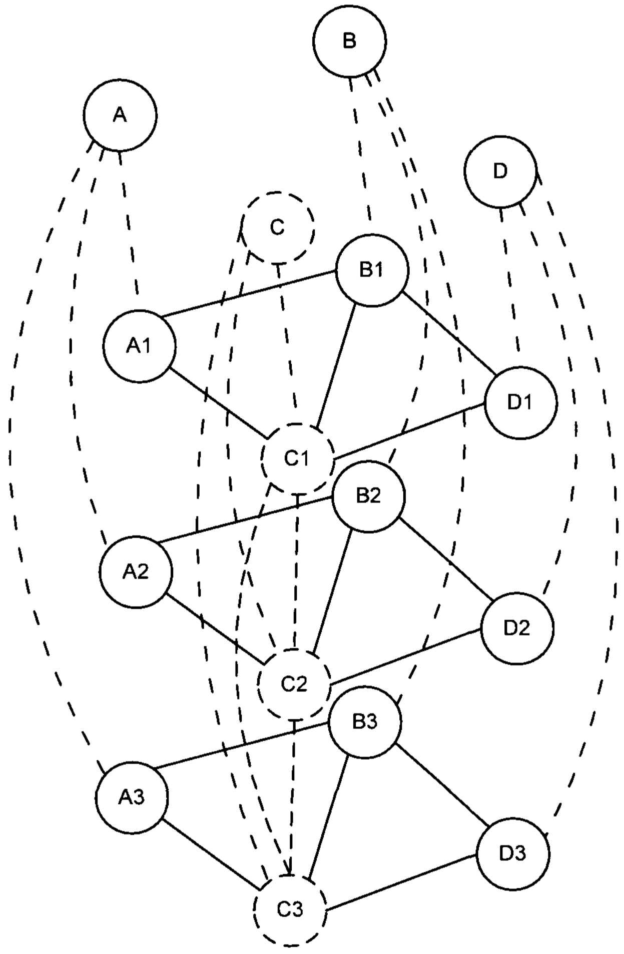

Figure 38: The final wavelength graph

8.2.2 Shortest Path Algorithm in Wavelength Graphs (SPAWG)

The Shortest Path Algorithm in Wavelength Graphs [21,31] is proposed for finding the minimal-cost path between two nodes in the wavelength graph. This algorithm makes use of the special structure of the wavelength graph and can reduce the complexity to $O((n+k)nk)$, where n is the total number of nodes and k is the total number of wavelengths in the network. This is lower than the complexity of the Dijkstra algorithm, which is $O((nk)^2)$.

Before defining the algorithm, we first present some notations. In the wavelength graph shown in Figure 38, each node, except a node in the additional layer, has two indexes: one is the row index (representing a wavelength) denoted by a number, the other is the column index denoted by a letter (representing a physical node). However, in the SPAWG algorithm, all the nodes of the graph are denoted by a single identifier. The nodes in the wavelength graph are labeled $1,\cdots,N$ in the following way: the nodes in the additional layer of the wavelength graph are labeled $1,\cdots,n$ and the nodes in the ith wavelength layer of the wavelength graph are labeled $ki+1,\cdots,ki+n$, where n is the total number of physical nodes, k is the total number of wavelengths in the network and $N=(k+1)n$. Thus, a node indexed h ($h > n$) is in row $\left\lfloor \frac{h}{n} \right\rfloor$ and column $h \bmod n$. The link cost of edge (i, j) is denoted by a_{ij}. If two nodes i and j are in the same column or row, this fact is denoted by $i \sim j$. The distance between the source node and node i is denoted by u_i. Two temporary variables R_i and C_j are associated with row i and column j, respectively. The temporary set and permanent set similar to the sets introduced in section 6.3 are denoted by T and P, respectively.

The SPAWG Algorithm is defined in the following. The original algorithm has several errors [31]. In the original algorithm, its steps 3 and 4 should be reversed. Also, an additional sub-step should be inserted to its step 2 to make sure the algorithm stops when the closest temporary node has infinite distance. We present here the corrected algorithm.

Step 1 initialization:

(1) $u_1 := 0$;

(2) If $i \sim 1$ then $u_i := a_{1i}$ else $u_i := \infty \, (\forall i)$

(3) $R_i := \min\{u_j : j \text{ is in the } i\text{th row}, j \neq 1\}, (\forall i)$

(4) $C_j := \min\{u_i : i \text{ is in the } j\text{th column}, i \neq 1\}, (\forall j)$

(5) $P := \{1\}; T := \{2, \cdots, N\}$.

Step 2 Designation of a New Permanent Label:

(1) Find the minimum of $R_i, C_j (\forall i, j)$.

(2) If the minimum found above is infinite, then STOP.

(3) Find an $h \in T$ with minimum u_h in the row or column which gave the minimum above (ties are broken arbitrarily).

(4) $T := T - \{h\}; P := P \cup \{h\}$

(5) If $T = \phi$ then STOP.

Step 3 Revision of Tentative Labels:

(1) If h, found in Step 2, is in row i and column j, then, for all $l \in T$ in row i and column j, set $u_l := \min\{u_l, u_h + a_{hl}\}$.

Step 4 Updating Row and Column Minimum:

(1) If h, found in Step 2, is in row i and column j,

(2) then $R_i := \min\{u_k : k \text{ is in the } i\text{th row}, k \in T\}$

(3) $C_j := \min\{u_k : k \text{ is in the } j\text{th column}, k \in T\}$ - Note: The minimum over an empty set is taken to be ∞ in (2) and (3).

(4) Go to Step 2.

8.3 RWA Schemes including Wavelength Conversion

8.3.1 The SPAWG Scheme

In the SPAWG scheme the original network is first transformed into the wavelength graph, and then the SPAWG algorithm is used to find the shortest path. The SPAWG scheme considers wavelength conversion and can find globally the shortest lightpath or semi-lightpath to accommodate the dynamic demands. This scheme needs to know the status of the entire network.

It can be seen that the complexity of the SPAWG scheme is very high. Thus, we propose two new schemes that also include wavelength conversion but have lower complexity. Instead of searching the entire network, the two schemes only search along some pre-constructed paths.

8.3.2 Shortest Path Routing with Wavelength Conversion (SPR-WC)

The SPR-WC scheme can be seen as a constrained SPAWG scheme. In this scheme, for each node pair a shortest path is first determined. When a traffic demand comes, the wavelength graph is partially generated. In this case, only nodes along the shortest path are used to construct the wavelength graph. As a result, in this graph the connection can only traverse links along the fixed shortest path, but it can make use of the wavelength converters at the intermediate nodes. After the generation of the partial wavelength graph, the SPAWG algorithm is used to find the shortest path to accommodate the traffic demand. For a given node pair, this scheme only needs the network status along the shortest path.

8.3.2 Disjoint Paths Routing with Wavelength Conversion (DPR-WC)

The DPR-WC scheme is similar to the SPR-WC scheme. The difference is that for each node pair it uses two link-disjoint paths to accommodate the traffic demands. When a traffic demand comes, nodes along the two paths are used to generate the constrained wavelength graph. Then the SPAWG algorithm is used to find the shortest path to accommodate the traffic demand.

8.4 Simulation Results

The wavelength conversion assumptions we made are similar to the assumptions in [20]. We set the wavelength conversion cost as follows. When the hop count is used, the wavelength conversion cost is set equal to 3 hops. When the Multiplier Guide is used, the wavelength conversion cost is set to be 3 times the channel usage cost. The channel usage cost is 25 in the 14-node NSFNET network and 20 in the 28-node Pan-European network. Both networks have 16 wavelengths. We assume there are 8 converters installed in each node.

8.4.1 Results for the SPAWG scheme

Figures 39 and 40 show the simulation results for the SPAWG scheme generated from the 14-node NSFNET network and the 28-node Pan-European network, respectively.. The results for the M_Dijkstra scheme generated from the same networks without wavelength converters are also shown in these figures for comparison. It can be seen that the SPAWG algorithm outperforms the M_Dijkstra algorithm because it makes use of the wavelength converters. However, the Multiplier Guide cannot improve the performance of the SPAWG algorithm.

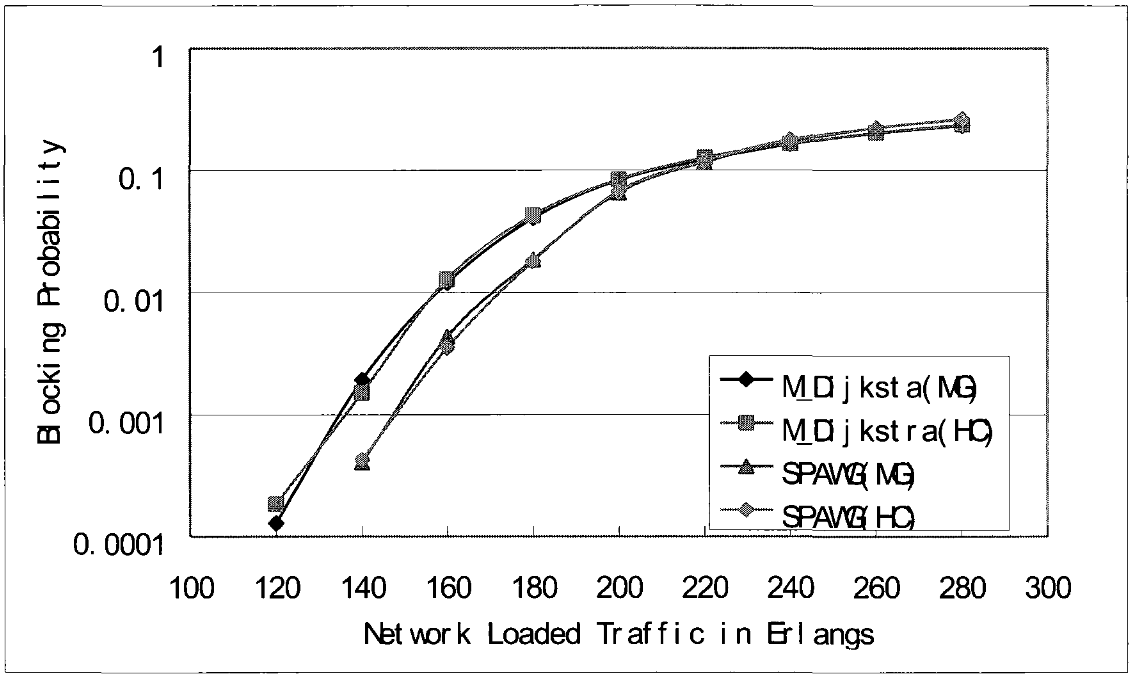

Figure 39: Results for the SPAWG scheme in the 14-node network

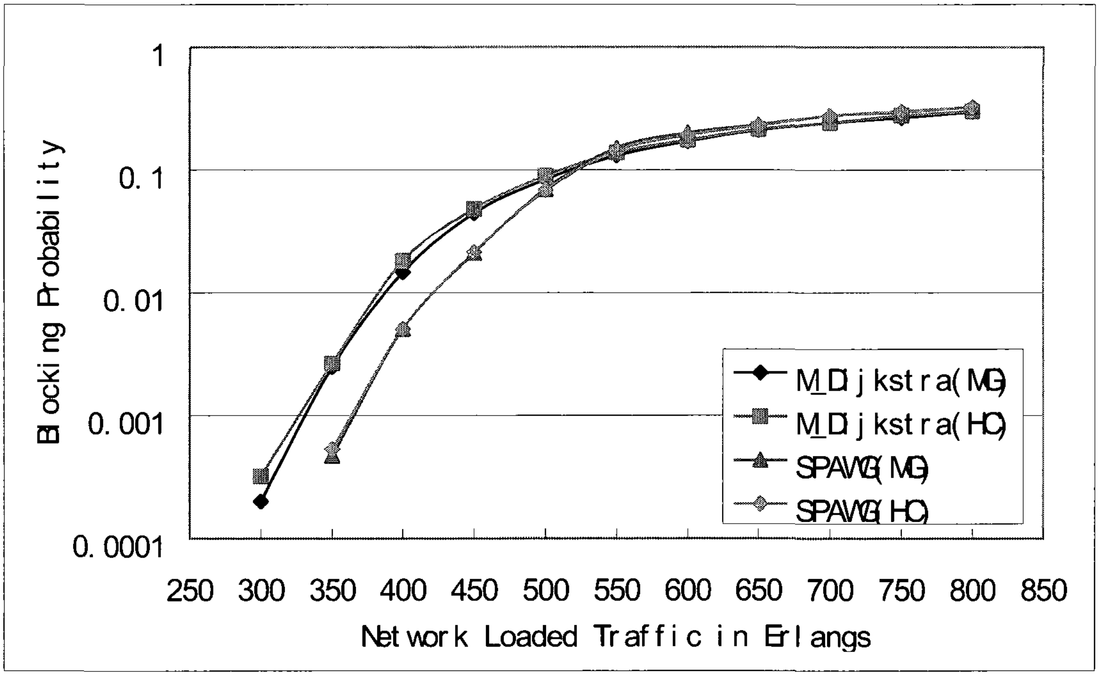

Figure 40: Results for the SPAWG scheme in the 28-node network

8.4.2 Results for the SPR-WC and DPR-WC schemes

Figures 41 and 42 show the simulation results for the SPR-WC and DPR-WC schemes generated from the 14-node NSFNET network and the 28-node Pan-European network, respectively. The results for the SPR-FF and LLR-FF schemes generated from the same networks without wavelength converters are also shown in these figures for comparison. It can be seen that, when both schemes use one path, the SPR-WC scheme outperforms the SPR-FF scheme. Also, when both schemes use two link-disjoint paths, the DPR-WC scheme outperforms the LLR-FF scheme. This is because the SPR-WC and DPR-WC schemes can make use of the wavelength converters. It can also be seen that when the Multiplier Guide is applied to the SPR-WC and DPR-WC schemes, their performance are improved.

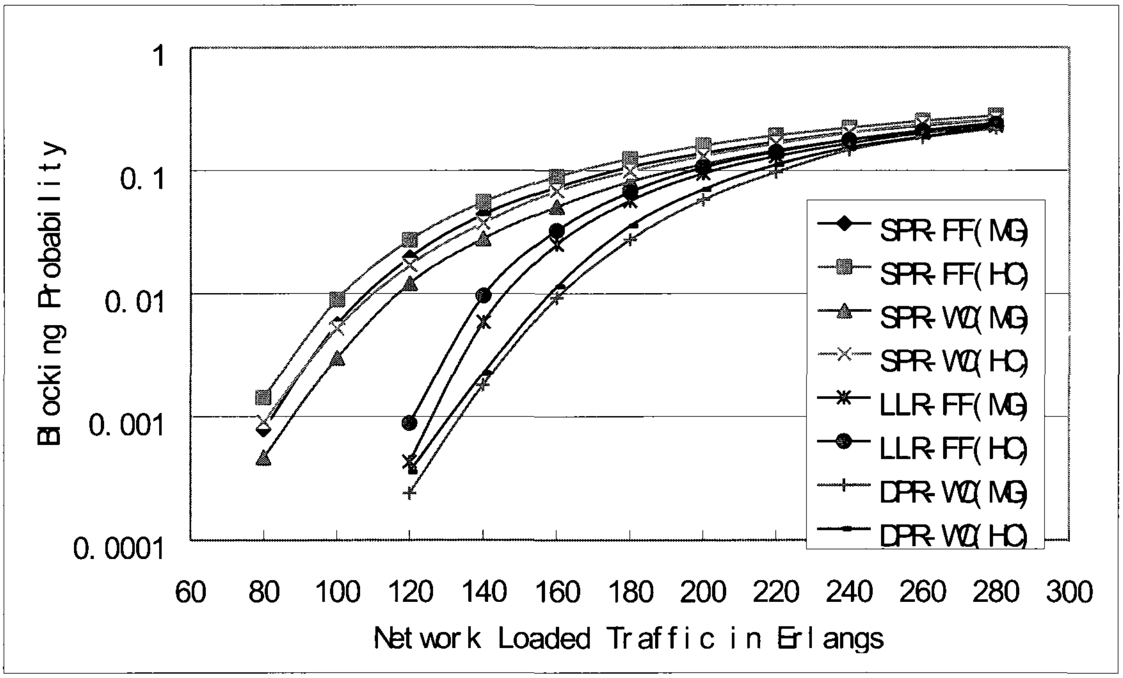

Figure 41: Results for the SPR-WC and DPR-WC schemes in the 14-node network

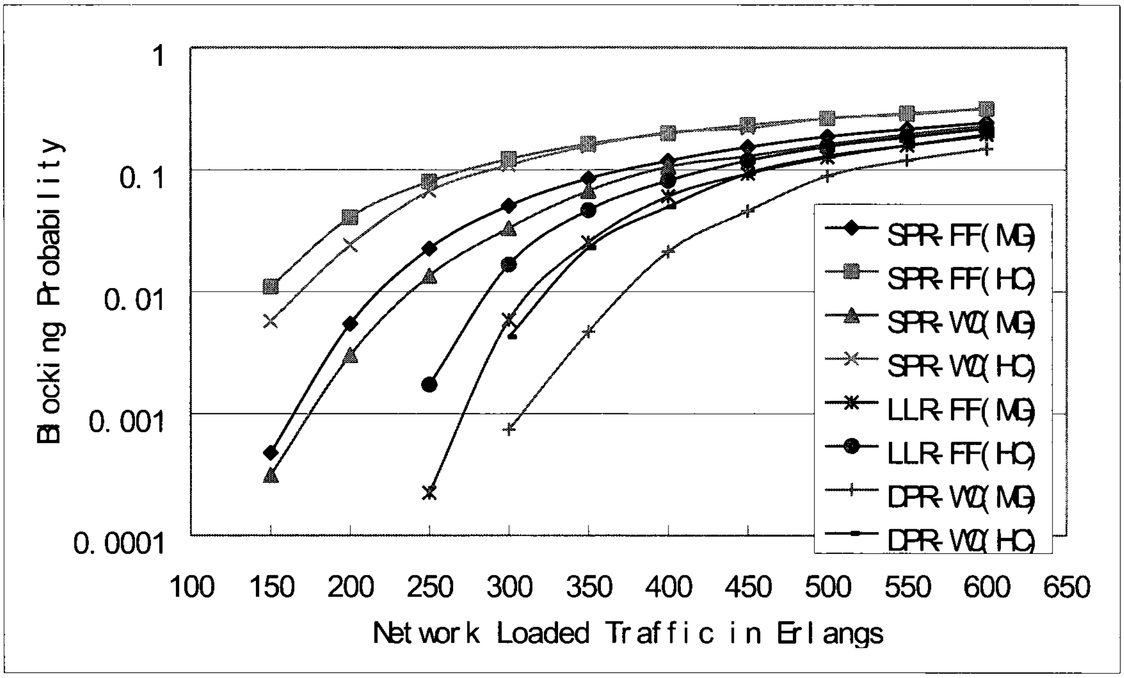

Figure 42: Results for the SPR-WC and DPR-WC schemes in the 28-node network

8.5 Conclusions

In this chapter we studied the RWA schemes that consider wavelength conversion. We implemented three schemes SPAWG, SPR-WC and DPR-WC. We compared them to three similar schemes M_Dijkstra, SPR-FF and LLR-FF without wavelength conversion. We found that the schemes that consider wavelength conversion perform better. Also the Multiplier Guide is applied to the SPAWG, SPR-WC and DPR-WC schemes. The result is that it cannot improve the performance of the SPAWG scheme, but it can improve the performance of the SPR-WC and DPR-WC schemes. The Multiplier Guide can improve the performance of the schemes that use a small number of paths for each node pair. When the number of paths increases, the improvement due to the Multiplier Guide becomes smaller. This trend can be seen in the KSP-FF scheme with different values of k. The SPAWG scheme searches the RWA solution in the entire network exhaustively, thus its performance cannot be improved by applying the Multiplier Guide. The SPR-WC and DPR-WC schemes only use one or two paths for each node pair. Thus, when the Multiplier

Guide is applied to these schemes, the performance can be improved

Chapter 9

Conclusions

In this thesis we studied the resource allocation problem in WDM networks under dynamic traffic demands. We focused on the question of how to improve the performance of dynamic RWA schemes by using information about the average distribution of the dynamic demands. We assume that the dynamic demands that come to the network obey a given distribution. From this distribution and the network topology, an optimized link cost for path computation, called the Multiplier Guide, can be computed. The value of the Multiplier Guide for each link takes into account the congestion status of that link. The Multiplier Guide can be used as the link cost to compute paths and improve network performance. In our work, we implemented several existing RWA schemes and applied the Multiplier Guide to these schemes. We found that the Multiplier Guide can improve the performance of some of these schemes.

9.1 Contributions of the Thesis

- We found that the Multiplier Guide can improve the following dynamic RWA schemes: SPR-FF, FAR-FF, LLR-FF, KSPR-FF, SPR-WC and DPR-WC.

- We found that the Multiplier Guide can bring better network utilization to the SPR-FF, FAR-FF and LLR-FF schemes.

- We found that the Multiplier Guide cannot improve the following dynamic RWA schemes: M_Dijkstra, SPAWG and ECSP-FF.

- We proposed six new dynamic RWA schemes: ULULC1, ULULC2, KSPR-FF, ECSP-FF, SPR-WC and DPR-WC.

- We detected some errors in the SPAWG and M_Dijkstra algorithms and submitted a correction to the related journal papers, which has been accepted and published.

9.2 Applying the Multiplier Guide to Real Network Applications

Our scheme can be applied to real networks. In real networks, the distribution of the dynamic demands can be obtained by measurements. We can measure the distribution of the dynamic traffic demands in real networks. Then we can use it to compute the Multiplier Guide. With the path computed by using the Multiplier Guide, the network performance is improved.

The advantage of the Multiplier Guide is that it is computed mathematically and no link utilization has to be obtained from real time network operations. This makes its implementation much easier. The schemes that use the link utilization as a factor to set the link cost have to measure the link utilization of the network in real time, which is hard to implement. Also, from our simulations it is shown that the schemes that use the Multiplier Guide as link cost perform as good as the schemes that use link utilization to decide link cost.

9.3 Future Work

The following items are proposed for future work:

In this thesis we have assumed that the distribution of the dynamic traffic demands is always fixed. In the real situation it may change from time to time. How to improve the performance of the dynamic RWA schemes in WDM networks with variable distributions of the dynamic traffic demand remains a topic to be studied.

In this thesis we have assumed that once the connection between two nodes has been established, it uses the same channel to transmit data until released. If the connections are allowed to be rerouted during the data transmitting process, the network performance can be improved. This is called network reconfiguration. How to improve the performance of dynamic RWA schemes in WDM networks with the consideration of network reconfiguration remains another topic to be studied.

References

[1] Y. Zhang, O. Yang, H. Liu, "A Lagrangean relaxation and subgradient framework for the routing and wavelength assignment problems in WDM networks", *IEEE Journal on Selected Areas in Communications*," Vol. 22, No. 9, pp. 1752-1765, November 2004

[2] T. Stern, K. Bala, "Multiwavelength optical networks: a layered approach," *Addison-Wesley Longman Publishing Co.*, 1999

[3] H. Zang, J. Jue, B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, 2000

[4] X. Chu, B. Li, "Dynamic routing and wavelength assignment in the presence of wavelength conversion for all-optical networks," *IEEE Transactions on Networking*, Vol. 13, No. 3, pp. 704-715, June 2005

[5] L. Li, A. Somani, "Dynamic wavelength routing using congestion and neighborhood information," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 5, pp. 779-786, October 1999

[6] Y. Yoo, S. Ahn, C. Kim, "Adaptive routing considering the number of available wavelengths in WDM networks," *IEEE Journal On selected Areas in Communications*, Vol. 21, No. 8, pp. 1263 – 1273, October 2003

[7] J. Zhang, J. Wu, G. Bochmann, M. Savoie, "Resource criticality analysis of static resource allocations and its applications in WDM network planning," *IEEE/OSA Journal of Optical Communications and Networking*, Vol. 1, No. 4, pp. 294–306, September 2009

[8] http://www.slimy.com/~steuard/teaching/tutorials/Lagrange.html

[9] R. A. Barry, P. A. Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers," *IEEE Journal on Selected area in communications*, Vol. 14, No. 5, pp. 852-857, June 1996

[10] S. Sybranabuanm, R. A. Barry, "Wavelength assignment in fixed routing WDM netowrks," *ICC* 1997

[11] R. Ramamurthy, B. Mukherjee, "Fixed-alternative routing and wavelength conversion in wavelength-routed optical networks," *IEEE/ACM Transactions on Networking*, Vol. 10, No. 3, pp. 351-367, June 2002

[12] E. Karasan, E. Ayanoglu, "Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks," *IEEE/ACM Transactions on Networking*, Vol. 6, No.2, pp.186-196, April 1998

[13] J. Kim, D. Lee, H. Sridhar, "Route-metric-based dynamic routing and wavelength assignment for multifber WDM networks," *IEEE Journal on Selected Area in Communications*, Vol. 24, No. 12, pp. 56-68, December 2006

[14] G. Duraes, A. Soares, J. Amazonas, W. Giozza, "The choice of the best among the routes in transparent optical networks," *Computer Networks*, Vol. 54, No. 14, pp. 2400-2409, October 2010

[15] A. Mokhtar, M. Azizoglu, "Adaptive wavelength routing in all-optical networks," *IEEE/ACM Transactions on Networking*, Vol. 6, No. 2, pp. 197-206, April 1998

[16] S. Xu, L. Li, S. Wang, "Dynamic routing and assignment of wavelength algorithms in multifiber wavelength division multiplexing networks," *IEEE Journal on Selected Area in Communications*, Vol. 18, No 10, pp. 2130-2137, October 2000

[17] E. Martins, M. Pascoal, J. Santos, "Deviation algorithms for ranking shortest paths," *International Journal of Foundations of Computer Science*, 1999

[18] J. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, Vol.17, No. 11, pp. 712-716, July 1971

[19] X. Chu, B. Li, I. Chlamtac, "Wavelength converter placement under different RWA algorithms in wavelength-routed all-optical networks," *IEEE Transactions on Communications*, Vol. 51, No. 4, pp. 607-617, April 2003

[20] K. Lee and V. Li, "A wavelength-convertible optical network", *Journal of Lightwave Technology*, Vol. 11, No. 5, pp.962-970, May-June 1993

[21] I. Chlamtac, A, Farago, T. Zhang, "Lightpath (wavelength) routing in large WDM networks," *IEEE Journal on Selected Area in Communications*,, Vol. 14, No. 5, pp. 909-913, June 1999

[22] A. Birman, "Computing approximate blocking probabilities for a class of all-optical networks," *IEEE Journal on Selected Area in Communications*, Vol. 14, No. 5, pp. 852-857, June 1996

[23] S. Subramaniam, M. Azizoglu, A. Somani, "All-optical networks with sparse wavelength conversion," *IEEE/ACM Transactions on Networking*, Vol. 4, No. 4, pp. 544-557, August 1996

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

[25] R. Ramaswami, K. Sivarajan, "Routing and wavelength assignment in all-optical

networks," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 5, pp. 489-500, October 1995

[26] A. Zapata-Beghelli, P. Bayvel, "Dynamic versus static wavelength-routed optical networks," *Journal of Lightwave Technology*, Vol. 26, No. 20, pp. 3403-3415, October 2008

[27] B. Ramamurthy, B. Mukherjee, "Wavelength conversion in WDM networking," *Journal on Selected Area in Communication* , Vol. 16, No. 7, pp. 1061-1073, September 1998

[28] A. Sridharan, K. Sivarajan, "Blocking in all-optical networks," *IEEE/ACM Transactions on Networking*, Vol. 12, No. 2, pp. 384-397, April 2004

[29] A. Ozdaglar, D. Bertsekas, "Routing and wavelength assignment in optical networks," *IEEE/ACM Transactions on Networking* ,Vol. 11, No. 2, pp. 259-272, April 2003

[30] D. Saha, "Lightpath versis semi-lightpath: some studies on optimal routing in WDM optical networks", *Photonic Network Communications*, Vol. 2, No. 2, pp. 151-161, 2000

[31] S. Yu, J. Wu, J. Zhang, G. Bochmann. "Corrections to "lightpath (wavelength) routing in large WDM networks" and "dynamic routing and assignment of wavelength algorithms in multifiber wavelength division multiplexing networks"," *IEEE Journal on Selected Areas in Communications*, Vol. 28, No. 9, pp. 1-2, December 2010

[32] D.P. Bertsekas, "Nonlinear programming," *Athena Scientific*, 1999.

Appendix

A Brief Introduction about How to Calculate the Multiplier Guide

A brief introduction about the mathematical approach used to calculate the Multiplier Guide is presented here. More details about this approach can be found in [1,7].

In the static RWA problem, given the network topology and the load demands, we have an objective function which has to be minimized under several constraints. The objective function is the sum of the penalties of the rejected demands and the network resource usage costs of the accommodated demands. The constraints of the problem are the wavelength channel capacity constraints, the demand flow continuity constraints, and the wavelength conversion constraints. The wavelength channel capacity constraints ensure the wavelength channel is used by no more than one traffic demand. The demand flow continuity constraints ensure that the accommodated demands are continuous along the paths. The wavelength conversion constraints ensure the accommodated demands satisfy the wavelength conversion assumptions.

The static RWA problem can be formulated as

$$\begin{aligned}&\text{Minimize}: f(x)\\&\text{Subject to}: g(x) \le 0, h(x) \le 0\end{aligned}$$

In the formulation, vector x represents an RWA solution. The $f(x)$ is a scalar function that represents the objective function. The $g(x)$ and $h(x)$ are vector functions: the $g(x) \le 0$ represents the wavelength channel capacity constraints while the $h(x) \le 0$ represents both the demand flow continuous constraints and the wavelength conversion constraints.

The original problem is hard to solve because the solution space is not convex. To solve the problem, Lagrangian Relaxation [31] is applied to relax the wavelength channel capacity constraints. With Lagrangian Relaxation, the problem is reformulated by adding the constraints multiplied by some non-negative variables, which are called Lagrangian Multipliers [31], to the objective function. The reformulated problem is shown as follows:

$$\begin{aligned}&\text{Minimize}: f(x)+\lambda g(x)\\&\text{Subject to}: h(x)\leq 0\end{aligned}\ .$$

Define the dual function [31] $q(\lambda)$ to be $q(\lambda)=\mathrm{Inf}_{x}(f(x)+\lambda g(x))$.

By definition, $q(\lambda)\leq(f(x)+\lambda g(x))$. Since $\lambda\geq 0$ and $g(x)\leq 0$, $f(x)+\lambda g(x)\leq f(x)$. Combine $q(\lambda)\leq(f(x)+\lambda g(x))$ and $(f(x)+\lambda g(x))\leq f(x)$, we have $q(\lambda)\leq f(x)$.

So we proved that, with any nonnegative vector λ, the minimized objective function value in the reformulated problem is always no greater than the minimized objective function value in the original problem.

When the dual function reaches its maximum and the primal function (the objective function in the original problem) reaches its minimum, the above inequality still holds, which is $q(\lambda^*)\leq f(x^*)$.

We want to find the multiplier vector λ^* that maximizes the optimal solution of the reformulated problem. This is the dual problem [31], which can be formulated as:

$$\begin{aligned}&\text{Maximize}: q(\lambda)=\mathrm{Inf}_{x}\,(f(x)+\lambda g(x))\\&\text{Subject to}: h(x)\leq 0,\lambda\geq 0\end{aligned}$$

If the optimal solution of the dual problem is found, it is an estimated lower bound of the original problem. This bound can be used to estimate how far a solution of the original problem is away from the optimal solution.

The dual problem is solved iteratively. Each time we get the multiplier vector $\bar{\lambda}$ first. Then the solution $\bar{x}$ that minimizes the reformulated problem is calculated. Knowing that the $\bar{x}$ may not be a feasible solution in the original problem, we apply a heuristic to generate a feasible solution $\bar{\bar{x}}$ of the original problem from $\bar{x}$. Then we compute the difference between $f(\bar{\bar{x}})$ and $f(\bar{x})+\bar{\lambda}(\bar{x})$. If the difference is small enough or the difference no longer decreases, we stop. Thus, we have obtained a near-optimal solution of the original problem. Otherwise we update the multiplier vector $\bar{\lambda}$ by using the subgradient method and start a new iteration.

In each iteration, with the updated multiplier vector, we want to minimize the objective function of the reformulated problem. In this problem, the wavelength channel capacity constraints are relaxed, so the demands can be seen independent of each other. Thus, the reformulated problem can be solved by finding the minimal cost path for every loaded traffic demand independently. In the reformulated problem, if a wavelength channel is used, the sum of its usage cost plus its Lagrange Multiplier is added to the objective function. As a result, the cost of each wavelength channel, which is used to compute minimal cost paths for the demands, is set to be its usage cost plus its Lagrange Multiplier.

A feasible solution of the original problem is generated from the solution of the reformulated problem by using a heuristic. In the feasible solution of the original problem, the demands traverse the same paths as they do in the solution of the reformulated problem. However, due to the wavelength channel capacity constraints, some demands that were accommodated in the reformulated problem may be rejected in the original problem.

When the iterations stop, a multiplier vector that maximizes the minimized objective function of the reformulated problem is finally found. With the multiplier vector, a solution that minimizes the reformulated problem is found next and finally a feasible solution of the original problem is found by applying a heuristic. Thus the static RWA problem is solved.

The multiplier vector obtained in the last iteration is used to develop the Multiplier Guide. It is found that the Lagrange Multipliers of all the wavelength channels in the same directional link are always the same. As a result, we can see the Lagrange Multiplier as a value associated with each directional link instead of each directional wavelength channel. From this mathematical approach we can see that, when the link costs are set to be their channel usage costs plus their Lagrange Multipliers obtained in the last iteration, and these costs are used to compute minimal cost paths to accommodate the demands, the static RWA problem is optimized. We assume that, since these link costs can optimize the static RWA problem, they can also optimize the dynamic RWA problem similar to the static RWA problem. Therefore we use the same link cost for the solution of the dynamic RWA problem, that is, the link cost is the channel usage cost plus the Lagrange Multiplier obtained in the last iteration. We call this link cost the Multiplier Guide.