ROUTING AND WAVELENGTH ASSIGNMENT IN ALL-OPTICAL NETWORK WITH MULTICAST TRAFFIC

BY

SUN YONG

A Thesis Presented to
The Hong Kong University of Science and Technology
In Partial Fulfillment
of the Requirements for
the Degree of Master of Philosophy
in Computer Science

Hong Kong, January 1998

Copyright © by Sun Yong 1998
**Authorisation**

I hereby declare that I am the sole author of the thesis.

I authorise The Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

---

I further authorise The Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

---

[Signature]

**Yong Sun**
ROUTING AND WAVELENGTH ASSIGNMENT IN ALL-OPTICAL NETWORK WITH MULTICAST TRAFFIC

BY

SUN YONG

APPROVED:

[Signature]

DR. JUN GU, SUPERVISOR

[Signature]

PROF. ROLAND T. CHIN, HEAD OF DEPARTMENT

Department of Computer Science

19 January 1998
ACKNOWLEDGEMENTS

First, I would like to take this opportunity to thank my supervisor Dr. Jun Gu for his advice and guidance during the whole period of my MPhil study. Without his assistance, this thesis could not be completed on time. I would also like to thank Dr. Bin Du for many helpful suggestions and discussions. In addition, I would like to thank Dr. Mounir Hamdi and Dr. Andrew Horner to be my thesis committee members.

During the period of working on the problem, many of my classmates has offered encouragement and support. They are Wang Lixin, Nong Ge, Hu Qinglong, Lau Ka Wo, Xu Jin and Xia Xuanyin. I want to take this opportunity to express my appreciation to them.

This thesis is dedicated to my parents.
3 SMT based multicast RWA algorithms
   3.1 Multicasting and its effect
   3.2 Support of multicasting
   3.3 A heuristic multicast RWA algorithm
   3.4 The SMT based multicast RWA algorithms
      3.4.1 The SMT based wavelength assignment algorithm
      3.4.2 The SMT based multicast routing algorithm

4 Performance evaluation
   4.1 Traffic model
   4.2 Effect of multicast traffic on the ideal switch
      4.2.1 The overall performance with multicast traffic
      4.2.2 Effect of multicast set size
      4.2.3 Effect of multicast load
   4.3 Performance of the SMT based RWA algorithms
      4.3.1 The overall performance of different multicast algorithms
      4.3.2 Effect of multicast set size
      4.3.3 Effects of multicast load

5 Conclusion and future work

Appendices

A K shortest paths problem
   A.1 Definition
   A.2 Martins’ algorithm

B Steiner Minimum Tree problem
   B.1 Definition
   B.2 The heuristic algorithms

References
LIST OF FIGURES

1. The low-loss region of an optical fiber.  
5. Fixed optical routing node.  
6. Dynamic optical routing node.  
7. A 24-node regional network.  
8. Effect of number of wavelengths with $T_1$.  
9. Effect of number of wavelengths with $T_2$.  
10. Effect of node capacity with $T_1$.  
11. Effect of node capacity with $T_2$.  
12. Part of Figure 10.  
13. Part of Figure 11.  
15. Effect of reconfiguration with 5 wavelengths, $n_c = 4$.  
16. Effect of reconfiguration with 5 wavelengths, $n_c = 5$.  
17. Two different ways to set up lightpaths.  
18. Two different lightpath settings.  
19. Performance of the ideal switch with multicast traffic ($T_1$).  
20. Performance of the ideal switch with multicast traffic ($T_2$).  
21. Performance of the ideal switch with various multicast set sizes ($T_1$).  
22. Performance of the ideal switch with various multicast set sizes ($T_2$).  
23. Performance of the ideal switch with various multicast loads ($T_1$).  
24. Performance of the ideal switch with various multicast loads ($T_2$).  
25. Performance of SMT based multicast algorithms with $T_1$, $n_c = 4$.  
26. Performance of SMT based multicast algorithms with $T_1$, $n_c = 5$.  

vii
Performance of SMT based multicast algorithms with $T_2$, $n_c = 4$. 49
Performance of SMT based multicast algorithms with $T_2$, $n_c = 5$. 50
Performance of SMT based multicast algorithms with $T_1$, $ml = 20\%$. 51
Performance of SMT based multicast algorithms with $T_2$, $ml = 20\%$. 51
Performance of SMT based multicast algorithms with $T_1$, $ml = 10\%$. 52
Performance of SMT based multicast algorithms with $T_2$, $ml = 10\%$. 52
The effect of various multicast set sizes with $T_1$, $n_c = 4$. 53
The effect of various multicast set sizes with $T_2$, $n_c = 4$. 53
The effects of multicast load with $T_1$, $n_c = 4$. 54
The effects of multicast load with $T_2$, $n_c = 4$. 54
The graph $G_1$. 60
The graph $G_2$. 61
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A heuristic wavelength assignment algorithm for CP1</td>
</tr>
<tr>
<td>2</td>
<td>The heuristic wavelength assignment algorithm.</td>
</tr>
<tr>
<td>3</td>
<td>A suboptimal routing algorithm</td>
</tr>
<tr>
<td>4</td>
<td>Traffic demand matrix $T_1$</td>
</tr>
<tr>
<td>5</td>
<td>Traffic demand matrix $T_2$</td>
</tr>
<tr>
<td>6</td>
<td>The heuristic multicasting routing algorithm.</td>
</tr>
<tr>
<td>7</td>
<td>The SMT based wavelength assignment algorithm.</td>
</tr>
<tr>
<td>8</td>
<td>A SMT based multicast routing algorithm.</td>
</tr>
<tr>
<td>9</td>
<td>Martins’ K shortest paths algorithm</td>
</tr>
<tr>
<td>10</td>
<td>The path deletion algorithm used in Martins’ algorithm</td>
</tr>
<tr>
<td>11</td>
<td>The MSTH algorithm</td>
</tr>
<tr>
<td>12</td>
<td>The Prim’s SMT problem</td>
</tr>
</tbody>
</table>
ROUTING AND WAVELENGTH ASSIGNMENT IN ALL-OPTICAL NETWORK WITH MULTICAST TRAFFIC

BY

SUN YONG

A Thesis Presented to
The Hong Kong University of Science and Technology
in Partial Fulfillment
of the Requirements for
the Degree of Master of Philosophy

Hong Kong, January 1998

ABSTRACT

All-optical wavelength-division-multiplexed (WDM) networks using wavelength routing are considered to be potential candidates for the next generation of wide-area backbone networks. The main problem in such networks is the Routing and Wavelength Assignment (RWA) problem, which has been proved NP-hard. Many algorithms are proposed in order to solve this problem efficiently. The main objective of the algorithms is to maximize the one-optical-hop traffic and balance the load on links as well as every intermediate nodes. However, most researchers assume that the traffic demand only contains unicast traffic.

Currently, multicasting is becoming a more and more important requirement in high-speed networks. Therefore, the traffic demands in the wavelength-routing network must be considered as hybrid, consisting of both unicast and multicast traffic. Although with some simple modifications, the current algorithms can be extended to support such a hybrid traffic model, the performance is not so good. The main problem is that there are many duplicated data transmitted in the network. In this thesis, we propose several SMT based algorithms which
can reduce the duplicated information using channel sharing technology. Finally, we compare the performance of these algorithms using simulation in a realistic network and some hybrid traffic models. The results illustrate that the SMT based algorithms can achieve remarkable performance under the hybrid traffic model.
CHAPTER 1

INTRODUCTION

1.1 High speed networks — demands and technologies

During the past 20 years, we have experienced three generations of networks which are based on the different underlying physical-level technology. Networks built before the emergence of fiber optic technology, i.e., which are based on copper-wire or microwave-radio technologies, are referred to as first-generation networks. Some examples of this generation of networks includes Ethernet, token bus, token ring, etc. The second-generation networks employ fibers in traditional architectures. One example of this generation is the upgrade of long-haul trunks in a wide area network (WAN) from copper or microwave-radio to fiber connections. Other examples are the Fiber Distributed Data Interface (FDDI), Distributed Queue Dual Bus (DQDB), and Broadband Integrated Service Digital Network (B-ISDN) etc. Improved performance can be achieved by employing the second generation of networks, because the fiber used in this generation of networks has some advantages such as higher data rates, lower error rates, and smaller electromagnetic emissions from the cabling etc. as compared to the copper-wire or microwave-radio employing in the first-generation. However, due to the electronic front ends employed at the network nodes, the bandwidth available in the second generation of networks can not exceed the peak electronic speed (a few giga bits per second). The third generation networks, which are often referred to as all-optical networks, employ totally new approaches to exploit the unique properties of fibers in order to avoid the electronic bottlenecks. In these networks, once the information enters the network, it will remain in the optical domain (and may not face the limitation of peak electronic speed) until it is delivered to its destination.
Currently, optical fiber has evolved to become the transmission medium of choice for high-speed communications. Today, single-mode fiber is widely used as the transmission medium for long-distance, point-to-point links, typically by telephone companies as the intercity trunks, including trans-Atlantic and trans-Pacific links. These links use the on-off keyed (OOK) intensity modulations of laser, and direct-detection receivers, where decisions are made simply on the basis of the amount of energy received during a bit duration. A single-mode fiber has two low-loss wavebands with approximately 30 THz bandwidth with over 200 nanometers between 1.3 μm and 1.5 μm (Figure 1). Using a modulation rate of 1b/Hz, the bandwidth mentioned above can be translated into nearly four orders of magnitude higher than the peak electronic data rate of a few giga bits per second.

![Figure 1: The low-loss region of an optical fiber.](image)

Nowadays, more and more people explore the internet everyday, seeking all types of information for work, business, and entertainment. The information includes not only simple text and data, but also images, audios, videos etc. New multimedia applications, such as high-definition television (HDTV), high quality digital audio, full motion video, and many other multimedia applications, have resulted in higher and higher bandwidth demands. With the above applications, it is envisioned that the future end-user will generate a sustained bandwidth
demand of approximately 1 Gb/s. With hundreds or even thousands of users, the overall network capacity would then have to be of the order of tera bits per second. According to the above discuss, we have seen that the potential bandwidth in the fiber is adequate to meet such a huge bandwidth demand. Therefore, the effective use of the bandwidth within the fiber becomes the main research problem.

Note that the bandwidth, which each end-user can access, is limited by the electronic speed of the equipment he/she uses. To exploit the huge bandwidth of lightwave networks, the key is to introduce concurrency among multiple-user transmissions into the network architectures and protocols. In an all-optical network, concurrency may be provided according to either wavelength or frequency (wavelength division multiple access—WDMA), time slots (time division multiple access—TDMA), or wave shape (spread spectrum or code division multiple access—CDMA). However, the basic need to have nodes synchronized to within one time slot (for TDMA) and one chip time (for CDMA) make these two technologies relatively less attractive than WDMA. WDMA, on the other hand, employs mostly the existing technologies associated with intensity-modulation direct-detection system, and is the current favorite of future network architectures because each end-user only operates at the bit rate of a wavelength division multiplex (WDM) channel while the user does not need to care about how many channels used and how many users are in the network.

The WDM lightwave networks can be classified into two categories: broadcast-and-select networks and wavelength-routing networks. In turn, each of these network architectures can use either single-hop approach or multihop approach. The single-hop approach refers to that the information, once transmitted as light, reaches its final destination directly without being converted into electronic form in between. The multihop approach, on the other hand, refers to that the information will be converted into electronic form at one or more intermediate nodes.
Figure 2: Topology of broadcast-and-select networks.

1.2 Broadcast-and-select networks

In broadcast-and-select architectures, the transmission from each station is broadcast to all of the stations connected by the network. The receiver extracts the desired one from all the signals. The topology of this type of network can be either a passive star or a passive bus (Figure 2). The passive star topology is preferred because of the efficient distributing of optical power and the fairness to the stations.

In this type of network, we can choose either fixed or tunable transmitters and receivers. For example, each station can use a fixed transmitter to transmit at an unique optical wavelength different from other stations. All transmissions are
broadcast to all stations, and at each station, a tunable optical filter select one of the wavelengths for reception. The synchronization between transmitters and receivers can be made by a median-access (MAC) protocol. The network fabric is totally passive (with the exception of optical amplifiers), consisting of optical couplers and splitters.

To improve the performance we need rapidly tunable transmitters or receivers, which are quite expensive. To get around this problem, we can use the multihop approach (Figure 3). For example, each station is provided with two fixed transmitters and two fixed receivers, while each transmitter is tuned to a different wavelength. Each station can transmit data directly only to those stations that have a receiver tuned to one of the wavelengths of its transmitters. With some routing algorithms, the information will be send to the destination crossing several hops. Thus, over the physical broadcast topology, there is a logical topology that determines the actual connectivity between the stations in the network. The disadvantage of this approach is that since information goes through multiple hops before reaching the destination, a significant portion of the network capacity is wasted.

Broadcast-and-select networks have several drawbacks. First, they require a large number of wavelengths, typically at least as many as the number of stations in the network, unless several stations are made to time-share a wavelength, that is, no wavelength reuse can occur in the network. As the number of channels increases, the requirements on the stability of transmitters and receivers become more stringent. Second, there is the so called splitting loss in this type of architecture. Since the transmitted power from a station is broadcast to all the stations in the network, each can only receive a small fraction of the transmitted power which decreases as the number of stations increases. Although optical amplifiers can be used to improve the power budget, the gain of the amplifier typically is not flat and the crosstalk of several WDM channels can not be avoided.

Broadcast-and-select networks are only suitable for LAN/MAN environments, and can not be extended to WANs directly, because of the drawbacks listed above. It is hard to imagine having a single star with many millions of users and millions
of wavelengths for a countrywide network. To construct such a network, the idea is to use the wavelength-routing architecture.

1.3 Wavelength-routing networks

The key idea behind wavelength-routing architectures is to use wavelength routing along with a limited amount of optical switch in order to be able to reuse the wavelength of the network. A typical wavelength-routing network is shown in Figure 4. A wavelength-routing network usually consists of many optical routing nodes, with some unidirectional fiber-links connecting two nodes. Some of the routing nodes have end-nodes attached to them. The end-nodes can be either end-users or another network. At any time each node may have logical connections with several, but not all, other nodes in the network. Each connection between a pair of nodes is carried on a certain wavelength. As long as the paths taken by two connections do not overlap, they can be on the same wavelength. For
example, in Figure 4, the connections from node A to node C, and node C to node E can be on the same wavelength $W_1$, while the connection from node B to node D will have to be on a different wavelength, $W_2$.

An optical routing node shown in Figure 4 has several input and output ports. Some of the ports are connected to other optical routing nodes, while others are connected to end-nodes. The routing nodes are able to route a signal carried on a wavelength coming in at an input port to an output port, independent of signals at other wavelengths. The routing node can be either fixed (Figure 5) or dynamic (Figure 6). In a fixed routing node, the routing is fixed, thus, the logical network is also fixed, while in a dynamic routing node, the routing can be changed, which also changes the logical network, in response to the network traffic pattern. Compared to the fixed one, the dynamic routing nodes have much more flexibility. Recently, an integrated-optic version of the node has been fabricated. Nowadays, several systems have been implemented as testbeds, where most of the implementations use dynamic routing nodes.
In both of these two types of routing nodes, the wavelength can not be changed. The signal which comes in carried on wavelength $W_1$ will still be carried on wavelength $W_1$ when it arrives at the output ports. Currently, the wavelength converters are also available. Obviously, a wavelength-routing network with wavelength converters is more flexible and has smaller blocking probability than one without wavelength converters. However, the wavelength converters are quite expensive, and the response time as well as the conversion range are still not satisfactory. Also, many researchers have obtained quantitative results showing that the performance difference of the systems with and without wavelength converters are not quite large, and as a result, the wavelength converters are not widely used.

Wavelength-routing networks are able to avoid the drawbacks of the broadcast-and-select networks. However, the benefits are obtained at the expense of hardware cost and complexity of network controls. Generally, the number of optical switches used in a routing node is no less than the number of wavelengths used in the network. A routing node with $n$ input ports and output ports and $M$ wavelengths requires $2n \times M$ multiplexers/demultiplexers and $M \times n \times n$ optical switches. To reduce the hardware cost, we often limit the number of wavelengths
used in the network. On the other hand, in order to support as much traffic as possible, we usually use the multihop approach. Therefore, the problem becomes how to assign the wavelengths and how to route the traffic in order to use these limited wavelengths efficiently. This problem, called Routing and Wavelength Assignment (RWA) problem, is the main focus in this thesis.

The remaining part of this thesis will be organized into three chapters: In the next chapter, we will first describe the RWA problem, then we will discuss a heuristic algorithm proposed by other researchers which can solve the RWA problem quite efficiently and can perform well in an unicast traffic environment. However, this algorithm can not be applied directly in a multicast traffic environment. In subsequent chapters, we will propose some improved algorithms which can provide multicasting support. Then we will apply these algorithms in a realistic network with a hybrid traffic model and compare the performance.
CHAPTER 2

ROUTING AND WAVELENGTH ASSIGNMENT PROBLEM

2.1 Dynamic and static RWA problems

As mentioned in the previous chapter, the main problem in a multihop wavelength-routing network is the Routing and Wavelength Assignment (RWA) problem. The main objective is to minimize the blocking probability of the connection requests for a given physical topology (fiber interconnection pattern) consisting of \( N \) routing nodes and \( P \) wavelengths, subject to a physical constraint that on each fiber link, no two connections use a common wavelength. This problem has been proved to be \( NP \)-hard.

The RWA problem can be classified into two classes of problems. One is dynamic RWA problem, which tries to find a path from the source to the destination by possibly reconfiguring the routing nodes in responding to a connection request. The other is static RWA problem, which first configure the routing nodes to construct a logical network according to some long term average traffic demand matrix \((t_{ij})\). Then a path is picked from the logical network to establish the connection requests. Both problems can be solved using either the single-hop approach, which refers to no end-nodes on the path, except for the source and the destination, or the multihop approach, where there are some end-nodes in the middle of the path. A single-hop virtual connection established between two nodes is often referred to as a lightpath, while the logical graph created by assigning lightpaths to some node pairs is referred to as a lightpath graph.

A similar routing problem arises in circuit-switched telephone networks. In this situation, we must route connections by selecting a path for each connection
such that there is a circuit available to accommodate the call in every link on the path. The difference between the routing in the circuit-switched networks and the one in the wavelength-routing networks is that in wavelength-routing networks, we not only need to satisfy the edge-disjoint constraint, which refers to the fact that two paths can not use the same wavelength on the same physical edges, but also need to satisfy the wavelength continuity constraint, which demands that the same wavelength must be assigned to all the segments of a lightpath. Notice that if we are allowed to use dynamic wavelength converters inside the optical network, then the wavelength-routing network becomes equivalent to the circuit-switched telephone network; therefore, we will use the term "circuit-switched” network to refer to both a circuit-switched telephone network and an optical network which uses the dynamic wavelength converters.

The dynamic RWA problem requires fast reconfigurable optical switches in order to reduce the setup time of the connection, while the static RWA approach has a much lower requirement. We will only consider the multihop static RWA problem in the remaining parts of the thesis.

The multihop static RWA problem can be solved with two steps, configuration of the routing nodes and the routing of the traffic. Since a connection request will be blocked when either the algorithm can not find a path or the end-nodes in the path can not process the connection. In a wavelength-routing network, the number of connections that the end-nodes can process is much less than the total number of connections established in the network; therefore, we need to minimize the average hop distance of the connections. The two steps of a static RWA algorithm can be defined as follows:

**Step 1:** The wavelength assignment algorithm refers to the approach which assign the wavelengths to some node pairs in order to maximize the one-optical-hop traffic.

**Step 2:** The routing algorithm refers to the connection routing schemes which will produce low blocking probability for the resulting logical graph obtained by applying the wavelength assignment algorithm on the physical network.
2.2 The wavelength assignment problem

The wavelength assignment problem can be defined as optimizing the throughput of the network by assigning the wavelengths to node pairs for a given long term average traffic demand matrix \((t_{ij})\), subject to both edge-disjoint constraint and wavelength continuity constraint.

The wavelength assignment problem can be treated as a special routing problem, where we are limited in finding the lightpaths to minimize the blocking probability with the knowledge of the whole traffic pattern.

With different traffic patterns, the wavelength assignment problem can be separated into two cases. One is the uniform traffic pattern, which refers to each pair of nodes having equal traffic demands. In this case, maximizing network throughput is same as maximizing the number of lightpaths established between different node pairs. The other is non-uniform traffic pattern, which refers to traffic demands that are randomly distributed. The main objective here is to support as many traffic demands as possible. We will only consider non-uniform traffic case in this thesis.

2.2.1 Mathematical formulation

Basically, the wavelength assignment problem is a constrained optimization problem \((CP)\) where the goal is to maximize the network throughput, subject to both edge-disjoint constraint and wavelength continuity constraint. To describe the constraints we need two matrices, the wavelength assignment matrix and the connection-link indication matrix. The wavelength assignment matrix is defined as follows:

\[
Z = (z_{ij}(k)),
\]

(2.1)

where

\[
z_{ij}(k) = \begin{cases} 
1 & \text{if node } i \text{ and } j \text{ are connected through wavelength } k \\
0 & \text{otherwise.}
\end{cases}
\]

(2.2)
The connection-link indication matrix is defined as follows:

\[ M = (m_{(i,j),(l,m)}), \]  

where

\[ m_{(i,j),(l,m)} = \begin{cases} 
1 & \text{if connection } (i, j) \text{ and } (l, m) \text{ use a common link} \\
0 & \text{otherwise.} 
\end{cases} \]  

(2.4)

Assume that each connection is established using a fixed path, and the capacity of each wavelength is \( C \), then the wavelength assignment problem can be expressed in terms of a constrained optimization problem, \( CP \), which can be formulated as follows:

\[ CP : \max \sum_{ij} \min(t_{ij}, \sum_{k=1}^{P} z_{ij}(k)C). \]  

(2.5)

subject to the constraints

\[ \sum_{m=1}^{N} z_{im}(k) \leq 1, \quad \text{and} \quad \sum_{m=1}^{N} z_{mj}(k) \leq 1, \quad k = 1, 2, \ldots, P, \quad i, j = 1, 2, \ldots, N, \]  

(2.6)

and

\[ m_{(i,j),(l,m)}(z_{ij}(k) + z_{lm}(k)) \leq 1, \quad k = 1, 2, \ldots, P, \]  

(2.7)

for all distinct pairs \( i, j \) and \( l, m \).

Equation 2.2 implies the wavelength continuity constraint. While for each wavelength, Equation 2.7 states that there should be at most one connection starting from node \( i \) and at most one connection terminating at node \( j \) using wavelength \( k \). Constraint 2.8 indicates that if connection \( (i, j) \) and connection \( (l, m) \) share a common physical link, \( z_{ij}(k) \) and \( z_{lm}(k) \) cannot equal to 1 at the same time, i.e., connection \( (i, j) \) and connection \( (l, m) \) cannot both be set up using wavelength \( k \).
From the above formulation, we can clearly see that the connection-link indication matrix plays an important role in the wavelength assignment problem. Different path setup strategies for the connections will result in different connection-link indication matrices. Richard A. Barry et al. [3] give out the effect of path length:

\[ P_b = (1 - (1 - \rho)^H)^P \]  

(2.8)

where \( P_b \) is the probability that a connection can not be established, \( \rho \) is the probability that a wavelength is used on a link, \( P \) is the total number of wavelengths, \( H \) is the length of the connection in terms of the number of physical links it pass through. From Equation 2.8, we can see that when the length of the connection increases, the probability that the connection can not be established also increases. Therefore, we select to set up connection \((i, j)\) with the shortest path from node \( i \) to \( j \). Another reason for choosing the shortest path to set up the connection is that after this connection is established, there remains more links for the establishment of other connections.

The problem \( CP \) is a nonlinear-integer problem, which can be difficult to solve when \( N \) and \( P \) are large. By introducing an additional continuous variables, we can convert this nonlinear problem into a mixed linear integer problem. Notice that if \( C \geq t_{ij} \), then

\[
\min(t_{ij}, \sum_{k=1}^{P} z_{ij}(k)C) = t_{ij} \sum_{k=1}^{P} z_{ij}(k).
\]

Let \( LT = \{ij|t_{ij} \leq C\} \), for each connection \( ij \notin LT \), let

\[
x_{ij} = \min(t_{ij}, \sum_{k=1}^{P} z_{ij}(k)C),
\]

then we have

\[
x_{ij} \leq t_{ij}, \text{ and } x_{ij} \leq \sum_{k=1}^{P} z_{ij}(k)C.
\]

The problem \( CP \) can then be converted into the following mixed integer linear
problem

\[ CP_{\text{mix}} : \max \left( \sum_{ij \in LT} t_{ij} \sum_{k=1}^{P} z_{ij}(k) + \sum_{ij \notin LT} x_{ij} \right) \] (2.9)

subject to

\[ \sum_{k=1}^{P} z_{ij}(k) \leq 1, \text{ if } ij \in LT, \] (2.10)

\[ x_{ij} \leq \sum_{k=1}^{P} z_{ij}(k), x_{ij} \leq t_{ij}, \text{ if } ij \notin LT, \] (2.11)

\[ \sum_{m=1}^{N} z_{im}(k) \leq 1, \quad \text{and} \quad \sum_{m=1}^{N} z_{mj}(k) \leq 1, \] (2.12)

\[ k = 1, 2, \ldots, P, \quad i, j = 1, 2, \ldots, N, \]

and

\[ m_{(i,j), (l,m)}(z_{ij}(k) + z_{im}(k)) \leq 1, \quad k = 1, 2, \ldots, P, \] (2.13)

for all distinct pairs \(i, j\) and \(l, m\).

Note: if \(t_{ij} \leq C\) for all the \(ij\)'s, then the problem is a binary linear problem.

Relaxing \(CP\) or \(CP_{\text{mix}}\) by ignoring the constraint that a connection \((i, j)\) must use the same wavelength, we get the channel assignment problem in a circuit-switched network, or the wavelength assignment problem in a wavelength-routing network with wavelength converter (prove can be found in [20]). If the number of wavelengths \(P\) is large enough, we can always find a lightpath between any two nodes \(i\) and \(j\), then the RWA problem becomes an ideal centralized switch with \(N\) inputs and \(N\) outputs which do not have any congestions inside the switch.

The problem \(CP\) and \(CP_{\text{mix}}\) can be approximately solved by repeat the assignment of one wavelength \(P\) times, with every time updating the traffic demands by removing the already assigned part. Since each time we only focus on a binary linear problem (\(CP1\)), it is much easier than the original one. The problem \(CP1\)
can be defined as follows:

\[ CP1 : \quad \max \sum_{ij} \min(t_{ij}, C)z_{ij} \quad (2.14) \]

where

\[ z_{ij} = \begin{cases} 
1 & \text{if the wavelength is assigned to connection } (i, j) \\
0 & \text{otherwise} 
\end{cases} \quad (2.15) \]

subject to

\[ \sum_{m=1}^{N} z_{im} \leq 1, \quad \text{and} \quad \sum_{m=1}^{N} z_{mj} \leq 1, \quad i, j = 1, 2, \ldots, N, \quad (2.16) \]

and

\[ m_{{(i)}}(z_{ij} + z_{im}) \leq 1, \quad (2.17) \]

for all distinct pairs \( i, j \) and \( l, m \).

Let \( z_{ij}^* \) be the optimal solution of \( CP1 \), Modify the traffic matrix \( T \) as

\[ \bar{t}_{ij} = \max(t_{ij} - Cz_{ij}^*, 0). \]

Solve \( CP1 \) with \( t_{ij} \) replaced by \( \bar{t}_{ij} \), the problem CP can be solved approximately by repeating the above procedure \( P \) times.

### 2.2.2 A heuristic algorithm

Even though \( CP1 \) is relatively easier to solve than problem \( CP \) or \( CP_{mix} \), it is still difficult when \( N \) is large. For example, for the 24-node network shown in Figure 7, the number of variables in \( CP1 \) is 552 and the number of constraints is 23822. Z. Zhang et al. [24] proposed a heuristic algorithm for an approximate solution to the problem \( CP1 \), which runs in polynomial time. The algorithm is shown in table 1.

Since the objective is to maximize the sum of one-optical-hop traffic, it is desirable that wavelength should be assigned to connections with the largest
traffic demands first. Then the algorithm removes all connections which share a common edge with the connection assigned. Find the connection which has not been removed and has the largest traffic demands, assign the wavelength to it. The procedure is repeated until all the connections are removed.

There are \( N(N - 1) \) possible connections. In the worst case, the algorithm has to check all the \( O(N^4) \) elements in the connection-link indication matrix, so the complexity of this algorithm for \( CP1 \) is \( O(N^4) \).

Repeat the algorithm \( P \) times with each time the assigned traffic demands removed, we can get an approximate solution of the problem \( CP \) or \( CP_{mix} \).

Notice that the lightpath graph created by this algorithm is not guaranteed to be connected. The approach to solve this is that after assigning \( P - 1 \) wavelengths, it is necessary to check whether the result graph is connected or not. If not, first connect all the unconnected subgraphs using the \( P \)-th wavelength to the shortest paths between the subgraphs, then assign the \( P \)-th wavelength to the connections that do not share a common edge with the path.

The whole wavelength assignment algorithm is shown in Table 2.
Step 1: Sort the traffics according to their demands in descending order.

Step 2: Generate the matrix $m$ as described above.

Step 3: Assign the wavelength to connection 1, set $k = 1$.

Step 4: Remove column $j$ with $m_{kj} = 1$ for $j = k + 1, \ldots, N$.

Step 5: Assign the wavelength to the first column (connection) not removed from the matrix in the $k$-th row, say connection $l$, $l > k$. If no such $l$ exists, goto step 7.

Step 6: Set $k = l$, goto step 4.

Step 7: Assign the wavelength to all the physical links that have not been assigned.

Table 1: A heuristic wavelength assignment algorithm for CP1

Step 1: Use the algorithm shown in table 1 to assign $P - 1$ wavelengths.

Step 2: For the result lightpath graph, check whether it is connected or not, if the lightpath graph is not connected, connect the unconnected subgraphs through the shortest paths using the $P$th wavelength.

Step 3: Remove those connections with common edge with the paths used in step 2 according to the connection indication matrix.

Step 4: Assign $P$th wavelength the remain item in the connection indication matrix.

Table 2: The heuristic wavelength assignment algorithm.

2.3 The routing problem

After setting up the lightpath graph, each user/node can have a virtual connection to any other user/node in the network. Each virtual connection travels along a sequence of lightpaths with intermediate access stations served as relay stations. The virtual connections are created and torn down by the network in response to instantaneous user-to-user connection requests. The goal of the routing algorithm is to choose a path for each of the connection requests with minimized blocking probability.

Routing algorithms have traditionally been divided into two classes: optimal
and suboptimal algorithms. A routing algorithm is said to be optimal if all the
possible combinations of input-output connections are considered, using all the
available information on the topology and the status of the network, and then
one path is selected according to an optimizing criterion. On the contrary, an
algorithm is said to be suboptimal if it makes choices based on the analysis of
a subset of the possible combinations. Clearly the Optimal algorithms should
perform better than suboptimal ones, but preliminary studies indicate that it
is possible to find suboptimal algorithms whose performance is only slightly de-
graded, but runs much faster.

Many researchers have proved that the blocking probability in such a routing
problem can be written as:

\[ P_b = 1 - ((1 - \rho^F)(1 - \rho_n^n))^H \]  (2.18)

where \( P_b \) is the blocking probability of a connection, \( H \) is the hop distance between
the source and the destination of the connection, \( \rho \) is the probability that a
connection passes through a hop, \( F \) is the total number of connections that can be
supported between two nodes of a hop, including multiple lightpaths established
between two nodes and multiple connections time sharing the same lightpath.
Note that since \( \rho F \) is the expected number of connections currently established
between two nodes, \( \rho \) is a measure of the load on the links of the network, \( \rho_n \) is
the probability that there is a connection pass through a node. \( n_c \) is the node
capacity which refers to the total number of connections which can be processed
by the node. Since \( \rho_n n_c \) is the expect number of connections crossing the node,
\( \rho_n \) is a measure of the load of all the input/output links of the node.

When given a network, \( F \) and \( n_c \) is fixed, the blocking probability is propor-
tional to \( \rho, \rho_n \) and \( H \). To minimize the blocking probability equals to minimize
\( \rho, \rho_n \) and \( H \). At a specific time, the connections established in the network are
fixed, to minimize the blocking probability of the future connection requests, we
need to try to balance the load on every hop and every node.

Currently, many routing schemes are proposed to solve the routing problem.
The most simple routing scheme is to assign the shortest path to the connection.
Because no network status is considered, the routing can be done very quickly. However, the performance is quite poor. An alternative scheme is to treat the path load as the weight of the link, then try to find a path with minimum weight to establish the connection. The performance of this scheme is much better than that of the previous scheme, but since the path load is varying with the time, every time we need to recalculate the minimum weighted path, the time for choosing paths, which is also the time used to set up a connection, will be quite long.

In a typical all-optical wavelength-routing network, the transmission time is quite short, which means the connection set up time can not be very long. Therefore, we use a suboptimal way to solve the routing problem, the algorithm is shown in Table 3.

**Step 1:** Calculate $k$ shortest paths (Appendix A) for any pair of nodes, put the result into a table.

**Step 2:** When a connection request comes, find the path can accommodate the connection and has the minimum load among these $k$ paths.

**Step 3:** If none of the paths in the $k$ sets is available, the connection is blocked. Otherwise, the connection is established using the path.

Table 3: A suboptimal routing algorithm

By only searching the path with minimum load in the $k$ shortest paths, we can approximately balance the load in the network while not making the searching time too long.

### 2.4 Numeric results

We simulate the algorithm stated above using a network shown in Figure 7. The nodes shown in the network are all routing nodes, where there are fixed set of end-nodes connected to these nodes (to keep the graph simple, no end-nodes are shown). Suppose the nodes in the network are connected by one bidirectional optical fiber while they are actually connected by two unidirectional fibers. The two long term average traffic demand matrices $T_1$ and $T_2$ used in the simulation
Table 4: Traffic demand matrix $T_1$

\[
\begin{pmatrix}
0 & 0 & 0 & 10 & 17 & 3 & 9 & 0 & 0 & 0 & 0 & 0 & 12 & 24 & 59 & 3 & 0 & 0 & 38 & 8 & 6 \\
0 & 0 & 32 & 17 & 12 & 17 & 3 & 14 & 15 & 0 & 0 & 0 & 0 & 15 & 15 & 31 & 3 & 0 & 0 & 20 & 0 & 0 \\
0 & 0 & 0 & 0 & 3 & 6 & 3 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 5 & 0 & 0 & 0 & 5 & 0 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 3 & 0 & 0 & 0 & 3 & 0 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

Table 5: Traffic demand matrix $T_2$

\[
\begin{pmatrix}
0 & 0 & 0 & 0 & 25 & 59 & 8 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 30 & 61 & 151 & 3 & 0 & 0 & 104 & 15 & 9 \\
0 & 0 & 90 & 45 & 24 & 41 & 9 & 28 & 31 & 0 & 0 & 0 & 0 & 0 & 35 & 35 & 53 & 3 & 4 & 0 & 49 & 0 & 0 \\
0 & 0 & 0 & 12 & 6 & 9 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 3 & 14 & 0 & 13 & 0 & 3 \\
0 & 0 & 0 & 0 & 6 & 5 & 7 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & 6 & 0 & 7 & 0 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

are shown in Tables 4 and 5 respectively, where $t_{ij}$ is the relative traffic demand between nodes $i$ and $j$ in the network. We use extensive simulation by generating calls randomly to obtain the call blocking probability.

2.4.1 Traffic model

Assume connections at node $i$ are generated independently of connections originating at other nodes according to a Poisson process with rate $\lambda \sum_{j=1}^{N} t_{ij}$. With probability $t_{ij} / \sum_{j=1}^{N} t_{ij}$, the connection is destined to node $j$. We assume that the
connection durations are independent and identically governed by an exponential distribution with mean $1/\mu$. All the connections are assumed to have same bandwidth $b$. For simplicity, assume that both $b$ and the capacity of a wavelength $C$ is measured by a common data rate, and the capacity of the end-nodes are some integer multiple of $C$. Define the connection granularity

$$g = \frac{\text{capacity per wavelength}}{\text{bandwidth required per connection}} = \frac{C}{b},$$

(2.19)

which is the number of connections supported on one wavelength by using TDM. Let $n_c$ denote the end-node capacity in units of $C$, where $1 \leq n_c \leq P$, define $n_{ij}$ to be the number of ongoing connections between node $i$ and $j$ at the moment when a new connection request arrives, The connection $(i, j)$ will be admitted if

$$1 + \sum_{j=1}^{N} n_{ij} \leq n_c g, \quad 1 + \sum_{i=1}^{N} n_{ij} \leq n_c g,$$

(2.20)

and a path with sufficient bandwidth can be found using the routing scheme.

We define the offered load to be

$$\rho = \frac{\lambda \max_{i} (\sum_{j=1}^{N} t_{ij})}{\mu}.$$  

(2.21)

The average connection blocking probability is defined by

$$P_{B_{ave}} = \frac{\text{total number of connections blocked}}{\text{total number of connections generated}}.$$  

(2.22)

For comparison, we consider the ideal centralized switch with $N$ inputs and $N$ outputs. A connection between node $i$ and $j$ will be admitted only if Equation 2.20 is satisfied. The centralized switch gives the lower bound of the call blocking probability. If $P$ is large enough, we can always find a lightpath to establish the connection, then the RWA problem is identical to this ideal centralized switch.

In the simulation, we first generate 1000 connection requests to ensure we get to the steady state, then we generate $10^6$ connection requests according to the traffic model stated above and collect the blocking probability as well as average hop distance. We choose to search 3 shortest paths for the routing of the connections in the simulation.
2.4.2 Effect of number of wavelengths

Figure 8: Effect of number of wavelengths with $T_1$.

Figures 8 and 9 show the effect of the different numbers of wavelengths used in the network. The figures are obtained by applying traffic demand matrices $T_1$ and $T_2$ to the network respectively. From the figures, we can see that both results determined using 4 wavelengths and 5 wavelengths are quite near to the result obtained by the ideal centralized switch. It is mainly because the heuristic algorithm mentioned above is approximately optimal, most connection requests are routed using one-optical-hop path. Although we can see that the average hop distance ($H_{ave}$ in the graphs) of the connection in the RWA case is about 30 percent larger than that in the ideal centralized switch case, the performance difference is still very small because the routing scheme we choose can balance the load quite well. We can also see that in both the two figures, the results obtained by using 4 wavelengths and 5 wavelengths are quite close, while using 5 wavelengths produced a slightly better performance. Generally, the performance of networks employing larger number of wavelengths will be better when com-
pared to those networks employing smaller number of wavelengths because as the number of wavelengths increases, more lightpaths can be established. But in our case, although the wavelength assignment is only approximately optimal, the routing algorithm works quite well, making the difference caused by adding a wavelength quite small.

2.4.3 Effect of node capacity

The effect of different node capacity are shown in Figures 10 and 11, which are created by applying traffic demand matrices $T_1$ and $T_2$ into the network respectively. From both of the figures, we can see that when node capacity $n_c$ increases, the blocking probability decreases significantly which verifies Equation 2.18. Also from the figures (for clarity, we zoom in to part of the figures and show the data in Figure 12 and 13 respectively), we can see clearly that the difference between the results of our RWA algorithm and the results of the ideal switch decreases when the node capacity increases. It is because our routing algorithm has balanced the
load on every lightpath quite well, the difference is mainly caused by blocking at the intermediate nodes, that is, the intermediate nodes can not process the connection request. Therefore, when the node capacity increases, the intermediate nodes can process more connection requests, thus, the difference decreases.

2.4.4 Effect of reconfiguration

Figures 14, 15 and 16 show the effects of reconfiguration. Above we have shown that the heuristic RWA algorithm can give near optimal results. Generally, since the optimization is based on the long term traffic pattern, when the traffic pattern changed, the result will not remain optimal, we need to reconfigure the lightpath graph. The effect of reconfiguration is quite large in Figure 14 which is created by using 4 wavelengths in the network. But in the other two figures which are created by using 5 wavelengths in the network, the effect of reconfiguration is quite small. This is mainly because when more wavelengths are used in the network, there are more lightpaths created. Although the optimal performance is almost not
Figure 11: Effect of node capacity with $T_2$.

affected, the network can tolerate more traffic changes. Also, we can see from the figures that, first, a different traffic pattern on the same lightpath graph will perform differently, with the more busty one performing worse. Second, since when the load gets higher, the blocking at the source and the destination become dominant, the effect of reconfiguration will become smaller.
Figure 12: Part of Figure 10.

Figure 13: Part of Figure 11.
Figure 14: Effect of reconfiguration with 4 wavelengths, $n_c = 4$.

Figure 15: Effect of reconfiguration with 5 wavelengths, $n_c = 4$. 
Figure 16: Effect of reconfiguration with 5 wavelengths, $n_c = 5$. 
CHAPTER 3

SMT BASED MULTICAST RWA ALGORITHMS

3.1 Multicasting and its effect

The heuristic RWA algorithm described in the previous chapter assumes that the connection requests are all unicast traffic. However, with a realistic network, because of the existence of multicast traffic, this may not necessarily be the case.

Current trends in networking applications indicate that there will be an increasing demand in future networks for multicasting—the capability of efficiently sending a stream of information from a single source to a number of destinations, $m$ (for $1 \leq m \leq N - 1$). Of course, the case $m = 1$ is the trivial unicast communication case, while the $m = N - 1$ case is the trivial broadcasting case, both of which have been well-studied in the past. Typical applications where multicasting is critical include the following:

1. the wire services used by news agencies, where a news agency such as the Associated Press need to distribute news reports from their bureaus to newspaper and radio stations throughout the world;

2. the multi-person video conferences, where every member need to send his opinion to all the other members simultaneously;

3. the video lectures, where a single speaker can address a large number of audiences;

4. the "multipoint-LAN interconnection", which allows large companies to treat their many geographically-distributed LAN's as a single large network;
5. the recently developed collaborative system, where any change made to the document must be sent to all the people working together; and

6. the "video on demand" services, where the service provider need to send a movie to some of its customer while another movie need to be send to some other customer at the same time.

Multicasting is also required in replicated and distributed databases to make updates. In addition, Multiprocessor systems require multicasting for cache coherency and message passing, too.

Multicasting can be classified either dynamically or statically depends on the destination set (often referred as the multicast set $M$) is created dynamically or statically. Once set up, a static multicast set remains unmodified until it is discarded. A connection to a static multicast set tends to be established according to a request from the source. Many multicast applications mainly deal with the static multicasting, for instance, the video lectures or video conferencing. Unlike the static multicast set, the membership of a dynamic multicast set can vary with time. The status of a dynamic multicast connection tends to be driven by destinations, which are either already part of the connection and wish to leave, or outside the connection and wish to join. The modification of the connection are usually made by the source according to a request from a destination. One example of an application which deal with the dynamic multicasting is entertainment video distribution.

In addition, the multicast set size of a multicast connection can also be varied, different connections contain different number of destinations. Typically, researcher can choose the set size $m$ to be either fixed, or varied following a certain distribution. For simplicity, in this thesis, we will only consider the cases of static multicast groups and fixed multicast set sizes.

Typically, the wavelength-routing networks are often designed as the backbones of some other network architectures, such as the telephone networks, ATM networks, Internet, etc. The traffic in the wavelength-routing networks usually should be considered as hybrid traffic, which consists of both multicast traffic
and unicast traffic. In the network which only concerned with whether the connection can be established or not, a multicast connection request must be set up simultaneously; that is, if a multicast connection request requires a connection between source $i$ and a set of destinations $M$ with set size $m$, and the path can not be found between node $i$ and a node $j \in M$ to accommodate the bandwidth required, then the whole connection is blocked. Therefore, a multicast connection needs at least $m$ hops to be established. According to Equation 2.18, the blocking probability increases when the offered load is same, as the average hop distance increases.

When the number of wavelength is larger enough, then we can always find a lightpath from the source to the destinations. In such an ideal case, which is an upper bound of the realistic networks, the Equation 2.18 can be rewritten as follows:

$$P_b = 1 - (1 - \rho_n n_c)^H,$$  \hspace{1cm} (3.1)

where $\rho_n$ is the load of a node, $n_c$ is the node capacity, $H$ is the hop distance of the connection, and $P_b$ is the probability that the connection will be blocked. Thus, for fixed traffic pattern and offered load, $\rho_n$ is fixed, when the hop distance increases, the blocking probability also increases.

If $m$ denotes the multicast set size, $ml$ denote the amount of multicast traffic in terms of the percentage of the total traffic demands, then the average hop distance of the connections, in the ideal case, can be calculated as:

$$H_{ave} = 1/((1 - ml) + ml/m).$$  \hspace{1cm} (3.2)

In the ideal case, the hop distance between any two nodes is 1, $1 - ml$ denotes the amount of unicast traffic and $ml/m$ denotes the amount of multicast connections (1 to $m$). From Equation 3.2, we can see that the more the multicast traffic, the larger the average hop distance. In addition, we can also see that when $m$ increases, the average hop distance of the connections also increases. However, when $m$ increases, the total number of connections decreases for fixed traffic pattern and offered load, therefore, the overall effect of various $m$ will be very small in the ideal case.
On the other hand, for a realistic network, we need to consider not only the blocking at the source and destination, but also the blocking at intermediate nodes and there may be some blocking are caused by no path can be found between the source and the destination. When \( m \) increases, a multicast connection may occupy more lightpath and pass through more intermediate nodes, thus, the blocking probability will increase.

### 3.2 Support of multicasting

Nowadays, the multicasting has become an important feature in high-speed networks. As mentioned above, the traffic in the wavelength-routing network often needs to be consider as hybrid. The network control system, e.g., the RWA algorithm, must provide efficient supports to these hybrid traffic demands.

Traditionally, point-to-point networks, including wavelength-routing networks, can provide multicasting support using either one of the following two approaches:

1. The multicast connection request is reated as several unicast connection requests which are related to each other by some constraints. If one unicast connection request can not satisfy these constraints, the whole multicast connection is blocked. One possible constraint might be that these unicast connection requests must be established together. If one connection can not be established, then all the connections are blocked. This approach is the simplest approach to support multicasting in a point-to-point network since only a very small part of network control system need to be modified. However, the performance is quite poor because there are many duplicated data will be send simultaneously through the same physical link.

2. The multicast connection request is treated as a single connection requests and the algorithm tries to find a path, typically a tree, to establish the connection. If such a tree can not be found to accommodate the connection, the connection is blocked. The advantage of using a tree is that it can efficiently reduce the duplicated information and enable parallel transmission
to the various destinations along the branches of the tree. However, using this approach to support multicasting requires great modification to the network control system. In addition, the complexity of routing algorithm will be increased if we want to find the exact optimal solution.

In the following sections, we will apply the first approach by modifying the heuristic RWA algorithm stated in the previous chapter to support the multicast traffic in the wavelength-routing network, second we will design two new algorithms using the second approach to improve the performance.

3.3 A heuristic multicast RWA algorithm

The simplest way to provide multicasting in a wavelength-routing network is to treat the multicast connection request as a set of unicast connection requests, where if one of the unicast connection can not be established, then the multicast connection request will be blocked; that is, if a connection request require to establish multicast connection from a source node \( s \) to some set of nodes \( M \), search a path from \( s \) to each of the nodes in \( M \) separately. If any of these paths can not be found or the nodes on the path can not process the connection, the connection request will be blocked, otherwise, the connection will be established using these paths.

The whole heuristic algorithm separated into the wavelength assignment part and the routing part, are shown in Table 2 and Table 6, respectively.

**Step 1:** Calculate \( k \) shortest paths for any pair of nodes, put the result into a table.

**Step 2:** When a connection request comes, for the source and each node in the destination set, find a path can accommodate the connection and has the minimum load among these \( k \) paths.

**Step 3:** If none of the paths in the \( k \) sets is available, the connection is blocked. Otherwise, the connection is established using the found paths.

Table 6: The heuristic multicasting routing algorithm.
Since the multicast connection requests are established by searching for a path from the source to every node in the destination set, it is desirable to carry the traffic on a lightpath. The objective of the multicast wavelength assignment algorithm is to maximize the one-optical-hop traffic between source-destination pairs, which is identical to the objective of the wavelength assignment algorithm when only the unicast traffic is considered. Therefore, the wavelength assignment algorithm used in the multicast traffic environment are the same as the algorithm used in the unicast traffic environment. However, since now the destination of a connection can contain more than one node, the routing algorithm used in the unicast traffic environment must be changed, which is made by searching for a path to accommodate the connection between the source and every node in the destination set. If any source-destination pair can not find such a path, the whole connection will be blocked.

3.4 The SMT based multicast RWA algorithms

(a) \((a,b), (a,c)\) and \((a,d)\) are connected by lightpath.

(b) \((a,b), (b,c)\) and \((b,d)\) are connected by lightpath.

Figure 17: Two different way to set up lightpaths.

The heuristic multicast RWA algorithm, although quite simple, works well with unicast traffic, but quite poor with multicast traffic. The main reason is that in both of the two parts of the algorithm, the wavelength assignment algorithm and
routing algorithm, with no consideration of multicast traffic has been undertaken. There are many duplicated data transmitted in the network, which waste many resources. Consider a small network, which contains 4 nodes \( \{a, b, c, d\} \) and the physical edges between them are connected as what is shown in Figure 17 using the thin lines. Assuming that only node \( a \) has some traffic demands to be send to the other 3 nodes, then the heuristic wavelength assignment algorithm will try to assign the wavelengths to connection \((a, b), (a, c)\) and \((a, d)\) (the algorithm should be able to connect both \((b, c)\) and \((b, d)\), although no traffic demands between them). This wavelength assignment scheme are shown in figure 17(a) using the thick lines. However, since in a multicast connection, the source send the same information to all the destination within the multicast set, the wavelength assignment scheme shown in Figure 17(b) may improve the performance since there is no duplicated information pass through physical link \((a, b)\), no matter what the destination is. Also if the lightpaths are established as shown in Figure 17(b), for connection request \( a \) to \( \{c, d\} \), duplicated information will be sent through \((a, b)\) using the heuristic routing algorithm. We need to find certain approaches to reduce the amount of duplicated information.

Since the RWA problem can be separated into two subproblems, the wavelength assignment problem and routing problem, we will try to consider the multicast traffic in each of these two problems in order to improve the performance.

3.4.1 The SMT based wavelength assignment algorithm

From Figure 17, we can see that if the lightpaths can be established in such a way that for a multicast connection \( a \) to \( \{c, d\} \), every maximum subtree without branches \(((a, b), (b, c)\) and \((c, d))\) of its Steinter Minimum Tree (SMT) (Discussed in Appendix B) can be constructed as a lightpath (Figure 17(b)), then the duplication of the data can be minimized. The basic idea of this algorithm is that given a long term traffic demand matrix \( t_{ij} \), the multicast load \( ml \), and the multicast set size \( m \), we can decompose \( t_{ij} \) into two parts \( t'_{ij} = t_{ij}(1 - ml) \) and \( t''_{ij} = t_{ij}ml \), the two parts are referred as the unicast demand and multicast demand in the following, respectively. For those nodes with nonzero multicast demand, we con-
struct a SMT for each multicast set using the algorithm described in Appendix B, then decompose the tree into maximum subtrees without branches, treat every subtree as a connection with some traffic demands. For example, in Figure 17, the connection \( a \) to \( \{c, d\} \) can be decomposed as connections \( (a, b) \), \( (b, c) \), and \( (c, d) \); the connection \( a \) to \( \{b, c\} \) will be decomposed as connections \( (a, b) \), \( (b, c) \). Add these connection to \( t_{ij}^{t} \), we can get a new traffic demand matrix \( tt_{ij} \). Applying this matrix to the original wavelength assignment algorithm, we can obtain the SMT based wavelength assignment algorithm with multicast traffic (Table 7).

**Step 1:** Generate the traffic matrix \( tt_{ij} \) as described above.

**Step 2:** Use the algorithm shown in Table 1 to assign \( P - 1 \) wavelengths.

**Step 3:** For the result lightpath graph, check whether it is connected or not, if the lightpath graph is not connected, connect the unconnected subgraphs through the shortest paths using the \( P \)th wavelength.

**Step 4:** Remove those connections with common edge with the paths used in step 3 according to the path indication matrix.

**Step 5:** Assign \( P \)th wavelength the remain item in the connection indication matrix.

Table 7: The SMT based wavelength assignment algorithm.

However, although the SMT based approach can establish more lightpaths, and efficiently reduce the duplicated information, the average hop distance of the connection is larger than the heuristic approach (Figure 17). In addition, an unicast connection request from \( a \) to \( c \) need to be connected as a path with hop distance equal to 2, this will reduce the unicast performance. Thus, this approach may achieve better performance in an environment with heavy multicast load, but in an environment with less multicast traffic, the performance may not be very good. In terms of average hop distance, the heuristic approach may be better. In addition, the information required in the heuristic wavelength assignment algorithm only contains the point-to-point traffic demands, which can be obtained by statistics. Thus, the heuristic approach may be more practical in a realistic network, compare to the SMT based approach.
3.4.2 The SMT based multicast routing algorithm

(a) \((a,b), (b,c)\) and \((c,d)\) are connected by lightpath.

(b) \((a,c), (a,d)\) and \((c,d)\) are connected by lightpath.

Figure 18: Two different lightpath settings.

After the establishment of the lightpath graph, we need to solve the routing problem such that the number of lightpaths occupied by a connection should be as small as possible, and the load on every lightpath and intermediate node should be balanced. If the lightpaths have been set up as what is shown in Figure 18(a), the heuristic routing scheme will send data through path \((a,b)\) twice, which is definitely not a good choice. Also if the lightpath graph of the sample network established as what is shown in Figure 18(b), the multicast connection from \(a\) to \(\{c,d\}\) may be established through path \((a,c),(c,d)\) for load balancing, while the heuristic routing scheme may only consider the path \((a,c),(a,d)\), path \((a,d,c),(a,d)\), or path \((a,c),(a,c,d)\), etc. So the routing of connections in lightpath graph can be improved if we construct a Steiner minimum tree (SMT) instead of just searching for the shortest path from the source to the destination.

Because the network status is varying with the time, traditionally, the Steiner Minimum Tree (SMT) problem need to be solved whenever a multicast connection request comes. Currently, the typical (and perhaps the fastest) suboptimal SMT algorithm gives out a approximately optimal solution within \(O(mn^2)\) time. The time to search for the SMT, which is also the time to establish a multicast
connection, may be quite long if the graph is large \((n \text{ large})\) and the multicast set size large \((m \text{ large})\). In an optical network, this will be even unacceptable because the transmission time usually is very short. To solve this problem, we develop a modified distance MST heuristic (MSTH) algorithm. Assuming that we have got the graph \(G = (V, E)\) and a multicast set of \(M \subseteq V\), the SMT based routing algorithm is shown in Table 8.

**Step 1:** Construct a new complete graph \(G' = (M, d)\) with any pair of vertices \(i, j \in M\) connected by \(k\) shortest paths from \(i\) to \(j\) in \(G\).

**Step 2:** When a multicast connection request comes, determine a minimum spanning tree \(T_1\) according to the current load on every link in \(G'\).

**Step 3:** Convert the edges in \(T_1\) to paths in \(G\) to form a subgraph \(G_s\).

**Step 4:** Find a minimum spanning tree \(T_s\) of \(G_s\).

**Step 5:** Construct a SMT \(T_H\) from \(T_s\) by deleting its edges so that all the leaves in \(T_H\) belong to \(M\).

**Step 6:** If the \(T_H\) can not accommodate the connection request, it will be blocked, otherwise, the connection will be established using \(T_H\).

Table 8: A SMT based multicast routing algorithm.

With the knowledge of the set size and traffic pattern, the algorithm can construct \(G'\) for all possible multicast set and store them in a table. This will considerably reduce the connection set up time since it is no longer necessary to check all the nodes in the network.

By combining the SMT based multicast routing algorithm with different wavelength assignment algorithm, we can get different schemes to solve the RWA problem with multicast traffic. The first scheme is obtained by combining the SMT based routing algorithm with the heuristic wavelength assignment algorithm, this approach, with both consideration on unicasting and multicasting, can work well in most combination of this two different types of traffic demands. Another scheme, which combines the SMT based routing algorithm with the SMT based wavelength assignment algorithm together, can work better in an environment where most connection requests are multicast connections. In next chapter, we will compare the performance of these schemes.
CHAPTER 4

PERFORMANCE EVALUATION

4.1 Traffic model

As mentioned before, the wavelength-routing networks need to provide support to hybrid traffic demands, consisting both unicast and multicast traffic demands. We still simulate the algorithms on the network shown in Figure 7. The traffic demand matrices used in the simulation are same as those shown in Table 4 and 5 without the connection independent assumption; that is, if \( m \) denotes the multicast set size, and there is a certain percentage which is referred to as multicast load \( ml \), of the connection requests require multicast connections, then the traffic demand between nodes \( i \) and \( j \) can be separated into two parts:

\[
t'_{ij} = t_{ij} \cdot (1 - ml), \quad \text{and} \quad (4.1)
\]
\[
t''_{ij} = t_{ij} \cdot ml \quad \text{(4.2)}
\]

where \( t'_{ij} \) is the uncasting traffic demand and \( t''_{ij} \) is the multicasting traffic demand.

At node \( i \), the uncasting connections will be generated according to a Poisson process with rate \( \lambda \sum_{j=1}^{N} t'_{ij} \). With probability \( t'_{ij} / \sum_{j=1}^{N} t'_{ij} \), the uncasting connection is destined to node \( j \). The multicasting connections start at node \( i \) will be generated according to a Poisson process with rate \( \lambda \sum_{j=1}^{N} t''_{ij} / m \). The \( m \) destinations will be randomly chosen from those nodes \( j \) with none zero \( t''_{ij} \) according to the value of \( t''_{ij} \). The durations of both unicast connections and multicast connections are all independent and identically governed by an exponential distribution with mean \( 1/\mu \). We use the same blocking model as used in Chapter 2 to decide whether a connection will be blocked or not, while the offered load will be defined
as:

\[ \rho = \frac{\lambda \max_i \left( \sum_{j=1}^{N} \left( t_{ij}^\prime + t_{ij}^\mu / m \right) \right)}{\mu} \]  

(4.3)

For comparison, we will still use the \( N \times N \) ideal centralized switch stated in the Chapter 2 with the assumption that the switch can also route the multicast connection from a node \( i \) to the multicast set \( M \) without congestion inside the switch. In the following sections, we will first show the effect of multicast connections on the ideal centralized switch, then we will apply the RWA algorithms we have obtained to evaluate the performance.

4.2 Effect of multicast traffic on the ideal switch

4.2.1 The overall performance with multicast traffic

![Blocking probability vs. offered load](image)

Figure 19: Performance of the ideal switch with multicast traffic \((T_1)\).

Figure 19 and Figure 20 illustrate the effect of multicast traffic with compare to the unicast traffic. The figures are created by applying traffic demand matrices
Figure 20: Performance of the ideal switch with multicast traffic ($T_2$).

$T_1$ and $T_2$ respectively. The notation $H_{ave}$ in the figures is defined as the average hop distance of the connection, which equals to 1 in the unicasting case. In the multicasting case, the average hop distance can be calculated as:

$$ah = \frac{1}{ml + ml/m} \quad (4.4)$$

which can be varied in the simulation because of the random generation of connection requests. From the figures, we can see that the multicast traffic will decrease the network performance heavily. According to the equation 3.1, we can also verify the results that with limited node capacity, when the average hop distance of the connection increases, the blocking probability will also increase. Since we are using the same traffic pattern for both unicasting and multicasting cases, it is clear that when the offered load increases, the difference between unicasting and multicasting will decrease.
Figure 21: Performance of the ideal switch with various multicast set sizes ($T_1$).

4.2.2 Effect of multicast set size

The effect of various multicast set sizes is shown in Figure 21 and Figure 22, which are created by applying traffic demand matrices $T_1$ and $T_2$ respectively. Notation $H_{\text{ave}}(k)$ used in the figures is the average hop distance of the connection with a multicast set size equal to $k$. From the figures, we can see that different multicast set sizes cause almost no difference. This is mainly because in the ideal centralized switch, a connection can only be blocked at source and destination nodes, no blocking will occur inside the switch. With the same traffic pattern, multicast load and offered load, the probability that a node can not process a connection will be the same, therefore, the blocking probability will be the same. Although as $m$ increases, the average hop distance increases, however, the total number of connections will decrease for the same traffic pattern. Thus, according to equation 3.1, the effect of different multicast set sizes will be quite small. In a realistic switch, this may not be the case since different multicast set sizes will affect the selection of the path.
4.2.3 Effect of multicast load

The effect of different multicast loads is shown in Figure 23 and Figure 24, which are obtained by applying the traffic demand matrices $T_1$ and $T_2$ respectively. From the figures we can clearly see that the blocking probability will become larger when the multicast load increases. Mainly it is because we have seen that the performance of the ideal switch with multicast traffic is worse than that with unicast traffic only. A larger multicast load means a larger multicast traffic demand, which will cause the network performance to decrease. The result can also be verified by considering the average hop distance. Since we are using same traffic pattern, it is desirable that the effect will become smaller when the offered load grows larger.
4.3 Performance of the SMT based RWA algorithms

Now we apply the multicast traffic into the wavelength-routing network, with different RWA algorithms. Some notations used in the following figures are explained: RWA\(_H\) refers to the heuristic multicast RWA algorithm, which consists of the heuristic wavelength assignment algorithm (table 2) and the heuristic multicast routing algorithm (table 6); RWA\(_{SMT}\) refers to the SMT based RWA algorithms, which contains two schemes. Scheme 1 includes the heuristic wavelength assignment algorithm as well as the SMT based multicast routing algorithm (table 8), and scheme 2 consists of the SMT based wavelength assignment algorithm (table 7) as well as the SMT based multicast routing algorithm.
4.3.1 The overall performance of different multicast algorithms

Figures 25 – 28 show the overall performance of the different multicast algorithms compared to the ideal centralized switch in an environment with 50% multicast traffic and the multicast set size equal to 3. We can clearly see that the performance of the heuristic multicast algorithm is quite poor when there are half connection requests which require multicast connections. The performance of both the SMT based algorithms are much better, when compared to the heuristic algorithm, although still not very good when compared to the performance of the ideal switch. The difference between the multicast algorithms and the ideal switch is mainly because the routing algorithm can not always find a small hop distance path for establish the connection. This can also be seen with the average hop distance. However, in the unicast case, although the average hop distance of the heuristic RWA algorithm is about 30% larger than that of the ideal switch, the algorithm still can work quite well because the routing algorithm can bal-
Figure 25: Performance of SMT based multicast algorithms with $T_1$, $n_c = 4$.

ance the load on every lightpath and intermediate node. On the other hand, because of the existence of multicast traffic, although the average hop distance of the heuristic multicast RWA algorithm is still about 30% larger than that of the ideal switch, the distance between the performance of these two cases is quite large because the heuristic multicast routing algorithm can not balance the load any more. SMT based multicast algorithm scheme 1, compared to the heuristic multicast algorithm, can only reduce the average hop distance for about 4 - 5%, the performance gain we obtained here is mainly because the SMT based routing algorithm can balance the load on lightpaths as well as on intermediate nodes much better than the heuristic one. The limitation of this scheme is that in the wavelength assignment part, no operation has been undertaken to reduce the duplicated information. The scheme 2, on the other hand, considers the reduction of the duplicated information in both the wavelength assignment part and the routing part. Thus, this scheme, although unable to reduce the hop distance, works best among these three algorithms in such a heavy multicast load environment.
Figure 26: Performance of SMT based multicast algorithms with $T_1$, $n_c = 5$.

However, when the multicast load decreases, scheme 2 is no longer the best choice among the three algorithms (Figures 29 – 32). Since the reduction of duplicated information is obtained at the expense of the increasing hop distance of unicast connections, scheme 2 can not deal with the environment where unicast traffic dominates. On the other hand, scheme 1 optimizes the lightpath for unicast traffic demands in the wavelength assignment part, and reduces duplicated information as well as balances the load in the routing part. Therefore, it can handle both unicast traffic and multicast traffic well. Also from the figures, we can see that the performance gain of the SMT based multicast algorithm, compared to the heuristic multicast algorithm, becomes smaller when the multicast load decreases. Basically, this is because the gain of the SMT based multicast algorithm is obtained by reducing duplicated information in the network, and balancing the multicast load. We will only get gain for multicast connection requests, for unicast traffic, the algorithms are identical, thus, no gain will be obtained. In the extreme case, if there are only unicast traffic demands, both
the SMT based multicast algorithms and the heuristic multicast algorithm are identical to the original heuristic RWA algorithm, which is quite near to the ideal centralized switch.

4.3.2 Effect of multicast set size

We have already seen that the effect of various multicast set sizes are quite small in the ideal switch. In a realistic network, this is definitely not the case. In the SMT based multicast algorithm, when the multicast set size increases, the blocking probability will also increase because when the multicast set size increases, it becomes harder for the routing algorithm to occupy the edge on the Steiner Minimum Tree (figure 33 and 34). The effects of various multicast set sizes will be reduced when the offered load increases since it is hard to find suitable paths for all the connection requests.
Figure 28: Performance of SMT based multicast algorithms with $T_2$, $n_c = 5$.

4.3.3 Effects of multicast load

In the previous section, we have seen that in the ideal switch case, a larger multicast load always make the blocking probability increase. The case is continued here. From Figures 35 and 36, we can see that when the multicast load increases, the average hop distance also increases, therefore, the blocking probability will also increase.
Figure 29: Performance of SMT based multicast algorithms with $T_1$, $ml = 20\%$.

Figure 30: Performance of SMT based multicast algorithms with $T_2$, $ml = 20\%$. 

51
Figure 31: Performance of SMT based multicast algorithms with $T_1$, $ml = 10\%$.

Figure 32: Performance of SMT based multicast algorithms with $T_2$, $ml = 10\%$. 
Figure 33: The effect of various multicast set sizes with $T_1$, $n_c = 4$.

Figure 34: The effect of various multicast set sizes with $T_2$, $n_c = 4$.  

53
Figure 35: The effects of multicast load with $T_1$, $n_c = 4$.

Figure 36: The effects of multicast load with $T_2$, $n_c = 4$. 

54
CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, we first reviewed the improvement of network technologies during the past 20 years. With the increasing of network bandwidth demands, wavelength-routing networks will probably become the backbone of the future Wide Area Networks (WAN). The main problem comes up with this network architecture is the hardware cost and the complexity of the network control, basically the RWA problem. In order to reduce the hardware cost, the number of wavelengths used can not be too large, thus, the multihop approach is required to maximize the network throughput. This will increase the complexity of the RWA problem. The RWA problem has been proved to be \emph{NP}-complete, the optimal solution is difficult to find. Although a simple heuristic RWA algorithm can be used to find a suboptimal solution and works quite well in the environment with only unicast traffic, it can not be applied to the environment with multicast traffic directly. With the consideration of the multicast traffic, first we proposed a simple modified algorithm which treats a multicast connection as several unicast connections and search path for each connection separately. This algorithm can be easily implemented, however, the performance is not so good. Then we proposed 2 SMT based algorithms to efficiently reduce the amount of duplicated information transmitted in the network. At the end we compared the performance of these algorithms in a network with a hybrid traffic model. From the results, we can see that the SMT based scheme 1 is the best because, first, the requirement is more practical for only requiring the information of the point-to-point traffic demands; and second, as compared to the SMT based scheme 2, unicast traffic can be handled better in scheme 1, while the unicast traffic may still be the main demands in future networks.

Our future work may include:
1. Improving the results of the wavelength assignment algorithm. Currently, the wavelength assignment algorithm used in this thesis is only approximately optimal. Although with a good routing scheme, the performance of the whole algorithm is quite reasonable in the unicast environment, but in a multicast environment, the results are still not good enough. The approximation is mainly made when the problem \( CP \) is decomposed into \( P \) iterations of problem \( CP1 \); using other models may achieve more accurate solutions. Some optimization techniques can also be employed to search for better solutions.

2. Evaluating the performance of the network in terms of packet switching, which is widely used in the current networking technologies. Most of the current works on wavelength-routing networks are carried out in terms of circuit switching. When changing to packet switching with delay and quality of service (QOS) constraints, the algorithm may need to be modified.
APPENDIX A

K SHORTEST PATHS PROBLEM

A.1 Definition

Let \( G = (V, E) \) be a directed graph, where \( V = \{1, 2, \ldots, n\} \) is a finite set whose elements are called vertices and

\[
E = \{e_1, e_2, \ldots, e_m\} = \{e_k = (i, j) | i, j \in V\}
\]

is a finite set whose elements are called edges. For each edge \( e_k = (i, j) \), \( i \) is called the tail vertex and \( j \) the head vertex of \( e_k \). In what follows, we assume that \( i \neq j \), for any edge \( (i, j) \in E \). Furthermore, for any vertex \( j \), let

\[
I(j) = \{(i, j) \in E | i \in V\}
\]

be the set of incoming edges of vertex \( j \); similarly for any vertex \( i \), let

\[
O(i) = \{(i, j) \in E | j \in V\}
\]

be the set of outgoing edges of vertex \( i \).

Let \( s \in V \) and \( t \in V \) be two distinct vertices of \( G = (V, E) \). A path \( p \) from \( s \) to \( t \) in \( G = (V, E) \) is a sequence of the form

\[
p = \{s = v_1, (v_1, v_2), v_2, \ldots, v_{k-1}, (v_{k-1}, v_k), v_k = t\}.
\]

where:

1. \( v_i \in V, \forall i \in \{1, \ldots, k\} \);

2. \( (v_i, v_{i+1}) \in E, \forall i \in \{1, \ldots, k - 1\} \).

A sub-path \( q \) of \( p \) is a path from \( v_i \in V \) and \( v_j \in V \) such that:
1. \( v_i, v_j \in V \cap p; \)

2. \( v_i \neq s \) or \( v_j \neq t; \)

3. \( q \) coincides with \( p \) just from \( v_i \) to \( v_j \).

Let \( P \) denote the set of all paths from \( s \) to \( t \) in \( G = (V, E) \). Let \( c_{ij} \) or \( c(i, j) \) be the weight of every edge \( (i, j) \in E \) and let \( c(p) = \sum_p c_{ij} \) be the weight of a path \( p \).

The shortest path problem is a classical graph problem, whose objective is the determination of a path \( p^* \in P \) for which \( c(p^*) \leq c(p) \) holds for any path \( p \in P \). This problem is well studied and hundreds of very efficient algorithm for this problem can be found in the literature.

The problem of determining the second, third, \ldots, \( k \)-th shortest path between a specified pair of vertices is also a well known graph problem. This problem can often be found in network problems where a path needs to be chosen subject to some complex set of constraints. This problem is also known as the ranking of the shortest path problem.

Currently, there are three classes of algorithm for solving this problem. The first class is based on the principle of optimality for the \( K \) shortest paths problem, which can be stated as any \( k \)-th shortest path is formed by \( j \)-th shortest sub-paths, where \( j \leq k \). The second class is based on the labelling shortest path algorithm. The last class is based on the path deletion concept due to Martins ([15]). The algorithm used in this thesis to determine \( K \) shortest paths is based on Martins’ algorithm.

### A.2 Martins’ algorithm

Martins’ algorithm is based on a very simple concept: once a shortest path is determined, a path deletion algorithm is used to remove it from the graph. This results in an enlarged graph where all the paths but the deleted one can be determined. So a sequence \( \{G_1, \cdots, G_k\} \) of graphs is defined, such that \( G_1 \) is
the given graph and its k-th shortest path is trivially determined from a shortest path of $G_k$.

KShortestPath(Graph G, Vertex i, Vertex j, int K)
/* computes the set \{p_1, \ldots, p_k\} of the k \leq K shortest paths */

k = 1;
while((k < K) \&\& (\pi(t) < \infty) {
    determine a shortest path p in G_1;
    k ++;
    delete p from G_k creating G_{k+1};
    determine a shortest path p in G_k;
    p_k is the corresponding path in G_1;
}

Table 9: Martins’ K shortest paths algorithm

Martins’ algorithm is shown in Table 9, where $\pi(t)$ is the shortest distance from s to t.

The path deletion algorithm simply duplicates some vertices on the path and changes some edges in the graph to delete the path. In order to allow the possible repetition of the initial and terminal vertices, the graph $G_1$ has to be augmented with a super-initial vertex s, a super-terminal node t, and zero weight edges (s, v_1) and (v_j, t). Assume

$$\tilde{p}_k = \{s, (s, v_1), v_1, \ldots, v_j, (v_j, t), t\}, \text{where } j \geq 2,$$

is the path to be deleted from some graph $G_k$. The path deletion algorithm is shown in Table 10:

An example of the algorithm is shown in Figure 37 and 38. In Figure 37, we show a graph $G_1$ where the super-initial vertex s and super-terminal vertex t has been added. In this graph, we can easily determine the shortest path $p_1 = \{s, 1, 2, 3, 6, t\}$ using any shortest path algorithm. In Figure 38, $p_1$ has been deleted and $G_1$ has been transferred into $G_2$ by adding some new vertices $\{1', 2', 3', 6'\}$ and update some edges. From $G_2$, we can see that the
1. The set \( \{v'_1, v'_2, \ldots, v'_j\} \) of new vertices is added to the graph.

2. The set of outgoing edges from node \( s \) is updated as

\[
O(s) \leftarrow O(s) - \{(s, v_1)\} \cup \{(s, v'_1)\}.
\]

3. The sets of outgoing edges from new vertices are defined as

\[
O(v'_k) \leftarrow \{v'_k, v\} | (v_k, v) \in O(v_k) \text{ and } v \neq v_{k+1}\} \cup \{v'_k, v'_k+1\}, \forall k \in \{1, \ldots, j-1\}
\]

and

\[
O(v'_j) \leftarrow \{(v'_j, v) | (v_j, v) \in O(v_j)\}.
\]

Table 10: The path deletion algorithm used in Martins' algorithm

![Graph G_1](image)

Figure 37: The graph \( G_1 \).

Path \( p = \{s, 1', 5, 2, 3, 6, t\} \) is the shortest path, therefore, converting the duplicated vertices to the original one, we can get the second shortest path in \( G_1 \) is

\( p_2 = \{s, 1, 5, 2, 3, 6, t\} \). Transfer \( G_2 \) to \( G_3 \) by deleting \( p \) from \( G_2 \), we can get \( p_3 \) by determine a shortest path \( p \) in \( G_3 \) then transfer \( p \) to \( p_3 \) by convert the duplicated vertices to the original one.

Martins' algorithm needs to run the shortest path algorithm \( K \) times and path deletion algorithm \( K - 1 \) times. In the worst case, all the shortest paths \( p_k \) contain all the vertices in the graph \( G_k \), for \( k = 1, 2, \ldots, K \), thus, in graph \( G_K = (V_K, E_K) \), we will have \( |V_K| = 2^{K-1}|V_1| \) vertices and \( |E_K| = 2^{K-1}|V_1| \) edges. The total complexity of Martins' algorithm will be \( O(2^K|V_1|^2) \). However,
in a more realistic analysis, e.g. in network problems, the weight of all the edges are positive, that is, the path is acyclic, the vertices in each path will not exceed $|V_1|$, the total complexity of Martins’ algorithm will be $O(K^3|V_1|^2)$. In practice, we also need only a very small $K$, typically $K \leq 3$. If $K$ can be treated as a constant number, the total complexity will be same as searching for the shortest path. Therefore, Martins’ algorithm is widely used because it only require a node’s outgoing edge, suitable for the traditional adjacency list representation of the graph.
APPENDIX B

STEINER MINIMUM TREE PROBLEM

B.1 Definition

Traditionally, most circuit-switched network architectures, such as telephone networks and ATM, support multicasting by using the multihop approach. To maximize the throughput of the network, the network control system needs to minimize the cost of the paths chosen for a multicast connection. The multicasting routing problem is equivalent to the construction of a Steiner tree. There are two reasons for basing efficient multicast routes on trees:

1. the data can be transmitted in parallel to the various destinations along the branches of the tree; and

2. a minimum number of copies of the data are transmitted, with duplication of data being necessary only at forks in the tree.

The goal is to minimize the weight of the Steiner tree, which is called the Steiner Minimum Tree (SMT) problem.

The SMT problem can be defined as:

Let \( G = (V, E, c) \) denote an undirected graph with \( V \) the set of vertices, \( E \) the set of undirected edges, and \( c \) a positive function on \( E \).

The SMT problem requires a connected subgraph \( T \) of \( G \) that includes a subset of the vertices \( M \subseteq V \) so as to minimize \( z = \sum_{(i,j) \in T} c_{ij} \).

The definition implies that \( T \) is a tree, which is acyclic. The vertices on \( T \) that do not belong to \( M \) is called Steiner vertices. The well-known special cases of the
SMT problem are "the shortest-path problem" for \(|M| = 2\) and "the minimum spanning tree (MST) problem" for \(M = V\). The corresponding problems in networking are unicasting and broadcasting, respectively.

The SMT problem is proved to be \(NP\)-complete, that is, it is only possible to find approximate solution using polynomial time algorithms. Many researchers have been working on this problem, and many heuristic algorithms with polynomial running time as well as several exact algorithms with exponential running time ([9, 10]) have been proposed.

### B.2 The heuristic algorithms

Most of the heuristic SMT algorithms are based on known graph algorithms, usually for the minimum spanning tree (MST) problem.

A well-known heuristic algorithm, which is called the distance MST heuristic (MSTH) algorithm proposed by Kou et al. ([13]) are shown in Table 11:

**Step 1:** Form the "distance graph" \((M, d)\), i.e., a complete graph with vertex set \(M\) and edge weights \(d_{kl}\), the shortest-path length in \(G\), for every \(k, l \in M\).

**Step 2:** Determine a spanning tree \(T_1\) on \((M, d)\) of minimum weight \(d(S)\).

**Step 3:** Convert the edges in \(T_1\) to paths in \(G\) to form a subgraph \(G_s\).

**Step 4:** Find a minimal spanning tree \(T_s\) of \(G_s\).

**Step 5:** Construct the SMT \(T_H\) from \(T_s\) by deleting its edges so that all the leaves in \(T_H\) are vertices in \(M\).

**Table 11:** The MSTH algorithm

Let \(L_{MSTH}\) and \(L_{OPT}\) be the length of the approximate SMT founded and the optimal SMT, respectively. It is know that \(L_{MSTH}/L_{OPT} \leq 2(1 - 1/t)\) where an optimal tree has \(t\) leaves. The MSTH runs in \(O(mn^2)\) time, with \(m = |M|\) and \(n = |V|\), where the construction of \((M, d)\) dominates.

Takahashi et al. ([22]) proposed a heuristic algorithm based on Prim's MST algorithm. Prim's algorithm begins with one singleton tree and iteratively con-
nects the nearest vertex to it, until there is one tree. This algorithm is shown in Table 12:

**Step 1:** Find a pair of vertices $i,j \in M$ that has minimum distance among all the distinct pairs in $M$, add all the vertices on $ij$ in to a set $N$.

**Step 2:** Choose one vertex $k \in M$ and $k \notin N$ which has the smallest distance to a vertex $l \in N$, add the vertices on $kl$ in to $N$.

**Step 3:** Repeat step 2 until all the vertex $k \in M$ has been add to $N$.

**Step 4:** Construct the SMT $T_H$ from the vertices in $N$.

Table 12: The Prim’s SMT problem

The performance bound of this algorithm is $L_{PRIM}/L_{OPT} < 2(1-1/m)$, where $L_{PRIM}$ is the length of the approximate SMT. The algorithm also runs in $O(mn^2)$ time. This algorithm is widely used in multicast networking problems because it can be used not only in static multicasting problems, but also in dynamic multicasting problems. That is, when a vertex $k$ needs to join the multicast group, we just need to find a vertex $l$ in the set $N$ such that $|kl|$ is minimized, then add all the vertices on $kl$ to $N$ and construct the new SMT $T'_{H}$. Using such an algorithm, we do not need to recalculate the whole SMT, therefore, the connection set up speed will be fast and the connection between the current member of the multicast group will not be affected at all.
REFERENCES


