2003

A heuristic for WDM path protection.

Min. Hou
University of Windsor

A HEURISTIC FOR WDM PATH PROTECTION

By
Min Hou

A Thesis
Submitted to the Faculty of Graduate Studies and Research
Through the School of Computer Science
In Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

May 2003

Your file Votre référence
ISBN: 0-612-82867-0
Our file Notre référence
ISBN: 0-612-82867-0

Abstract

Optical fibers and Wavelength Division Multiplexing (WDM) are being researched as well as commercially deployed as technologies that can satisfy the bandwidth requirements of the Internet today and the foreseeable future. Since optical resources are expensive, we need to develop network design mechanisms, which can achieve an efficient utilization of network resources in a reasonable computational time. In WDM networks, the failure of a single fiber link may cause very large data loss if the traffic is not rerouted quickly. Therefore, the survivability of optical connections has become a very important issue for WDM network design.

In this thesis, we present a heuristic method to determine the logical topology and routing scheme with WDM shared-path protection. In this protection scheme, a primary path and a backup path are determined for each optical connection. The simulations of our heuristic show that it is simple and efficient, and can be used for designing fault-tolerant logical topologies for practical-sized WDM networks.

Keywords: WDM, Wavelength Routed Networks, Physical Topology, Logical Topology, Single-Hop, Multi-Hop, Routing and Wavelength Assignment (RWA), Survivability, Protection, Restoration, MILP, Heuristic.

Acknowledgement

I like to take this opportunity to thank my advisor Dr. Arunita Jaekel and Dr. Subir Bandyopadhyay for their continuous support and valuable advice. Without their instructive guidance, this work would not have been achieved. I also like to thank Dr. Xiang Chen and Dr. Xiaojun Chen for their time and valuable suggestions.

Special thanks to the entire optical network group for providing me with help during the writing of this thesis.

I like to thank my family for their love and support which have always come in abundant.

Table of Contents

Topic	Page

List of Figures

List of Tables

List of Abbreviations

1. ATM - Asynchronous Transfer Mode
2. CPLEX - C Programming Language + simpEX
3. EURONET- European Optical Network
4. ILP - Integer Linear Programming
5. IP – Internet Protocol
6. LAN - Local Area Network
7. LP - Linear Programming
8. MILP - Mixed-Integer Linear Programming
9. NSFNET - National Science Foundation Network
10. OXC - Optical Cross-Connect
11. RWA - Routing & Wavelength Assignment
12. SDH - Synchronous Digital Hierarchy
13. WAN - Wide Area Network
14. WDM - Wavelength Division Multiplexing
15. WRN - Wavelength Routed Networks

Chapter 1: Introduction

Nowadays optical fibers and Wavelength Division Multiplexing (WDM) [Gre92, Muk97] are being researched and deployed as technologies that can satisfy the bandwidth requirements of the Internet today and the future. Optical fibers are the most attractive medium in today's telecommunication traffic, because of its high bandwidth, low error rate and low loss [Coc95]. Wavelength Division Multiplexing (WDM) technology can utilize the tremendous bandwidth of optical fibers by transmitting data over separate wavelength channels on each fiber. Each channel can operate at an electronic speed of a few gigabits per second. Transmissions on multiple WDM carrier wavelengths from different end-users are combined and transmitted on the same fiber [Muk97, RS98].

In WDM networks, Wavelength Routed Networks (WRN) are potential candidates for the next generation of wide-area backbone networks [Muk97]. In WRN, optical signals can be selectively routed, based on wavelength. The physical topology of a WRN consists of end nodes, optical routers and fiber links connecting them. Each end node has a limited number of optical transmitters and receivers. Each fiber link is capable of carrying a certain number of carrier wavelengths. End nodes may communicate with one another via all-optical paths, which may span multiple fiber links and router nodes. The optical routers used at the intermediate nodes can route the signals on incoming fibers to outgoing fibers as needed without any opto-electronic conversion. Such all-optical paths are referred to as lightpaths [CGK92, RS98]. Each lightpath can be viewed as a point-to-point connection from a transmitter at a source node to a receiver at a destination node.

The set of lightpaths presented in an optical network define a logical topology [Muk97] over the physical topology. The nodes in a logical topology are the end-nodes in the physical topology. Each edge in the logical topology is a lightpath, which provides a single-hop communication between two end nodes. The higher layer services are carried by the logical topology. An arbitrary logical topology may be defined using any given physical fiber network. In such an overlay architecture, the logical topology makes the physical topology invisible to the higher layer networks (such as SDH, ATM, or IP) [MPPM02].

1.1 Motivation

Ideally, we would like to set up lightpaths between all the $N(N\text{-}1)$ pairs for a N nodes network. However, this is usually not possible because the number of wavelengths available per fiber is limited and the optical transmitters and receivers at the source and destination nodes are expensive. The logical topology design problem is to determine how the lightpaths should be defined in order to make optimum use of network resources.

Without wavelength conversion [Joe93], a lightpath must use the same wavelength on all the links it traverses. This is called wavelength continuity constraint [Bra90], which is assumed in this study. Given a wavelength routed network, for all requests for communication, finding routes for the lightpaths through the physical topology and assigning wavelengths to these lightpaths is called routing and wavelength assignment problem (RWA) [RS95, ZJM00]. Logical topology design is actually a problem of selecting lightpaths and RWA.

When it is not possible to establish a direct lightpath between a source-destination pair, they still can communicate by a sequence of lightpaths through intermediate nodes. This is called multi-hop approach [Muk97, Bra93]. In our study, if we cannot find a single-hop path for a source-destination pair, we try to find a multi-hop path for it.

With different lightpaths assignments, the same physical topology can have many different logical topologies. The performance of each logical topology can be quite different. The performance can be evaluated by the network congestion, which is defined as the maximum traffic load on any lightpath. Therefore, the purpose of most studies is to design the optimal logical topology, i.e., the logical topology with the lowest congestion [LA91, RS98].

The adoption of the WDM technology makes it possible that a single fiber link can carry much more traffic than before. However, the failure of the fiber link in a WDM network may possibly affect a terabit of traffic per second. Therefore, the issue of survivability of WDM networks has become a very important problem [Fum99, GR00, MN01, AMM00, CPTD98]. There are two main approaches for fault management: restoration [RM99a, IG00, LDS98, DDHH+99, DG00] and protection [RM99b, CB98, EHS00, SG00].

In restoration [RM99a], the spare capacity is not reserved in advance. After the fault's occurrence, the available spare capacity has to be found and used to recover the disrupted traffic. If a fiber fails, the nodes adjacent to the failed fiber link send link-fail message to all the source-destination nodes of all connections that traverse the failed link. When a source node of a connection receives a link-fail message, it initiates a restoration-path search on a certain set of wavelengths. The restoration-path search is performed on each wavelength in parallel. If a restoration path is found, the connection is set up on the restoration path. If more than one restoration path is found for a connection, the first one found is utilized, and the others are released.

In protection [RM99b], spare capacity is reserved in advance. When we define our logical topology and decide that there has to be a logical edge from node i to node j, we have to set up two lightpaths from node i to node j. We will call the first lightpath as the primary lightpath and the second lightpath as the backup lightpath. In the absence of any fault in the network, all communication will use the primary lightpath only. When there is a fault in the network, a number of primary lightpaths which use the faulty component of the network cannot be used. In each case where the primary lightpath cannot be used, we use the corresponding backup path.

The vast majority of faults is due to a fiber cut and our main concern in this thesis is to handle fiber cuts. To make sure that a single fault due to a fiber cut cannot affect both the primary lightpath and the backup lightpath of a node pair, we have to make sure that the primary lightpath and the backup lightpath are edge-disjoint.

Recently, a number of researchers have proposed strategies to design fault tolerant logical topology in WDM networks. Some approaches formulate the problem as a Mixed Integer Linear Programming (MILP) [IL01]. The solution of the MILP formulation provides an optimal logical topology and an optimal scheme for routing the traffic between every source-destination pair. A complete MILP formulation for determining the optimal logical topology and routing scheme with WDM path protection is presented by Hong Guan [Gua03]. This formulation guarantees the lowest congestion. However, to solve these MILP formulations takes a very long time. In [Gua03], it has been reported that it takes several days to get the solution for a four node network. Therefore, the MILP formulation is infeasible even for small networks although it can provide optimal solution for the logical design problem.

1.2 Problem Outline

Since the process of solving the MILP formulation is too time consuming [HABJ+02], we developed a heuristic for determining a fault tolerant logical topology and the routing scheme over the logical topology. This heuristic can be solved much faster than the MILP formulation. For example, to find a fault tolerant logical topology for a 14 node network, it takes less than one second. Therefore, this heuristic is feasible for much larger networks. However, this approach speeds up the solution time at the cost of optimality.

For designing the logical topology, the physical topology, the characteristics of the fiber and the traffic matrix are major factors. The physical topology tells us the number of nodes (end nodes and router nodes) as well as the connectivity between the nodes. For each end node, the number of transmitters and receivers for each carrier wavelength is also known. The characteristics of the fiber tell us the number of wavelengths that can be carried by each fiber. In this work, we considered a static traffic scenario. In this case, the long-term average traffic for each node pair in a network can be conveniently represented by a matrix T which is usually called the traffic matrix [Muk97]. The $(i, j)^{th}$ element t_{ij} of the traffic matrix is the amount of traffic between the source-destination pair (i, j).

Given a physical topology and a traffic matrix T, we are required to find a logical topology. Over this logical topology, each traffic request between a source-destination pair can be accommodated by one or more lightpaths. That is, each request for traffic between the source-destination pair (i, j) can be handled by one or more single-hop path(s) from node i to node j and/or one or more multi-hop path(s) from node i to node j. This logical topology also provides protection against a single fiber failure, which, as mentioned earlier, is the predominant form of failures in optical networks. Since we use WDM path protection scheme, for each communication request, we need to determine the logical topology. For each edge in the logical topology, we need to find a primary and a backup route and the wavelengths for the primary and the backup lightpaths. In addition, we need to ensure that there is a transmitter at the source node and a receiver at the destination node for each lightpath. When we route the traffic, we try to route it by existing lightpaths first. If we cannot do so, we route it by adding new lightpaths.

1.3 Thesis Organization

The thesis is organized as follows. Chapter 2 provides a literature review of the basic terminology and techniques of WDM networks and fault management techniques for WDM networks. Chapter 3 starts with the MILP formulation of the logical topology design and the routing problem which has been studied in detail by [Gua03]. Chapter 3 also includes an overview of the heuristic algorithm for designing the logical topology which is the main topic of this thesis. Chapter 4 explains the implementation of the heuristic. Chapter 5 analyzes the results of experiments for some well-known networks. Chapter 6 concludes the thesis with a critical summary, and future work.

Chapter 2: Background Review

With the tremendous growth of the Internet and the World Wide Web (WWW), the data network is evolving to include more and more bandwidth-intensive network applications, such as data browsing on the WWW, java applications, video conferencing, etc. Optical networking [Gre96, Muk97] is the key technology to satisfy the increasing bandwidth requirements of the emerging applications. Optical fiber is the main medium in today's networks, because of its huge bandwidth, low error rate, reliability, and maintainability [Muk97, RS98].

In optical networks, in order to make use of the huge bandwidth available in the fiber, concurrency among multiple user transmissions can be achieved through wavelength using wavelength division multiplexing (WDM) [Muk97, RS98]. WDM is perceived to be a promising candidate to address the bandwidth shortage on the Internet. Survivability of WDM networks has grown into an issue of great importance, and is being researched by many researchers [Wu92, Ger98]. In this chapter, we give a brief review of background information related to the remainder of this thesis. WDM networks, wavelength routed networks, single-hop and multi-hop networks, logical topology design and network survivability are introduced in this chapter.

2.1 WDM Networks

Wavelength division multiplexing divides the huge bandwidth of a fiber (up to 50T bits/s) into many non-overlapping wavelength ranges, called WDM channels. Each WDM channel may be operated at a very high speed, e.g., peak electronic speed of a few gigabits per second (Gbps). Transmissions on multiple WDM channels from different end-users are multiplexed on the same fiber. In this way, the aggregate network capacity can reach the number of channels times the capacity of each channel. The number of distinct wavelengths that are simultaneously carried by a given fiber depends on fiber characteristics and other technological constraints, such as crosstalk (the effect of other signals on the desired signal).

A WDM system uses a wavelength multiplexer at the source node to combine several wavelengths on to a single fiber and separates the composite signal at the destination node using a wavelength demultiplexer (Figure 2.1).

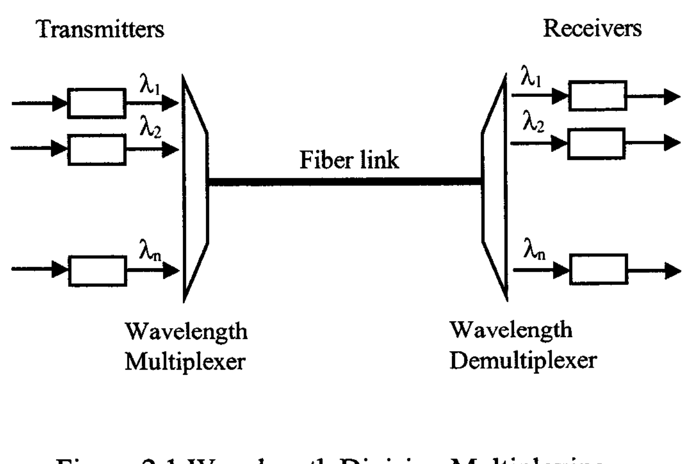


Figure 2.1 Wavelength Division Multiplexing

2.2 Wavelength Routed Networks

The commonly used architectural form for WDM networks is Wavelength Routed Networks (WRN) [Muk97], which is the network architecture in our research. Wavelength Routed Networks (Figure 2.2) are composed of one or more wavelength selective nodes called optical routers and fibers interconnecting these nodes. Each optical router has a number of input and output ports. These ports are connected to either end-nodes or other optical routers. The optical routers could be Optical Cross-Connects (OXCs), Add-Drop Multiplexers, etc.

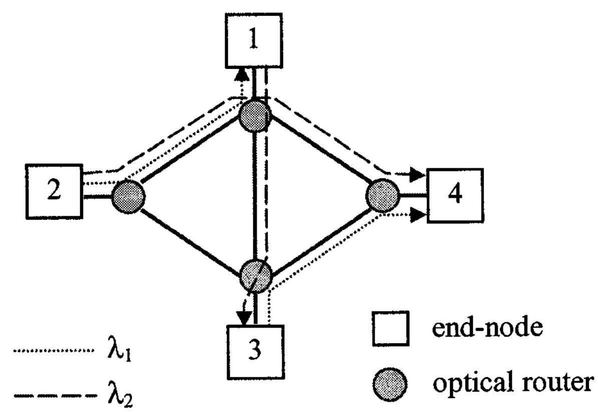


Figure 2.2 Wavelength Routed Networks (Physical Topology)

Each optical router makes its routing decision according to the port and the wavelength of the input signal coming to that port. All the routing is done in the optical domain. Signals routed to the same output port should have different carriers. In Figure 2.3, signals on λ_1, λ_4 can be routed to the same output port, same as signals on λ_2, λ_3. Signals on a number of input fibers may be routed to different output fibers.

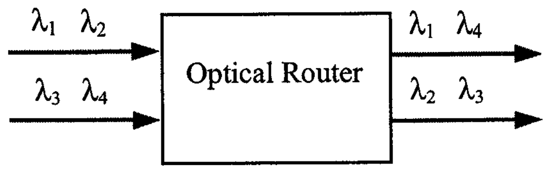


Figure 2.3 Optical Router

Depending on the design and the components in use, an optical router may have a variety of capabilities. For example, its routing matrix may be static or reconfigurable, it may provide wavelength conversion or not. These features have direct influence on the operation and the scalability of the network.

2.3 Physical Topology and Logical Topology

2.3.1 Physical Topology

The physical topology of a wavelength routed network consists of end-nodes, optical routers and fibers connecting them [Muk97]. End-nodes are the nodes can generate, send and receive data. End-nodes are attached to the optical routers. An example of a physical topology is given in Figure 2.2. The physical topology is the network topology seen by the optical layer. In this topology, end-to-end connections between the end-nodes are established on wavelength channels through one or more optical routers. Such an end-to-end connection behaves like a high-speed transparent pipe between the end-nodes, which is called a lightpath [Muk97]. For example, in Figure 2.2, the lightpath between node 2 and node 4 is an end-to-end connection on wavelength λ_2 through three optical routers.

2.3.2 Lightpath

A lightpath uses a channel on each link in a path between the source and the destination. Depending on the capabilities of the network, the lightpath could be set up or taken down upon request of the higher layer. This could be viewed as a circuit-switched service, similar to the service provided by older generation telephone network. Alternatively, the network may provide only permanent lightpaths, which are set up at the time the network is deployed.

In wavelength routed networks, if an optical router is equipped with a wavelength converter [Joe93], a lightpath may switch between different wavelengths on its route from its source to its destination. Otherwise, the lightpath should be assigned to the same wavelength along its route, this is called the wavelength continuity constraint [Bra90]. In Figure 2.2, the lightpath between $2 \rightarrow 4$ spans two fiber links. But it is assigned same wavelength λ_2 along its route. Due to the high cost of wavelength converters, most proposed optical networks do not allow wavelength conversion. In this thesis, we assume the wavelength continuity constraint.

In addition, lightpaths sharing a common fiber should be assigned to different wavelengths. However, if two lightpaths do not share a fiber on their routes, they can be assigned to the same wavelength. In Figure 2.2, lightpaths between the end-nodes $1 \rightarrow 3$ and $2 \rightarrow 4$ (similarly $2 \rightarrow 1$ and $3 \rightarrow 4$) are assigned to the same wavelength because they do not share any fiber on their routes. Therefore, wavelengths reuse can be achieved in wavelength routed networks. This feature results in a tremendous reduction in the number of wavelengths required for building the wide area networks.

2.3.3 Logical Topology

The logical topology [Muk97] is the network topology seen by the higher layer. Figure 2.4 is the logical topology for the physical topology in Figure 2.2. The nodes in this topology correspond to the end nodes of the physical topology, and each directed edge in this topology represents a lightpath has been set up in the physical topology.

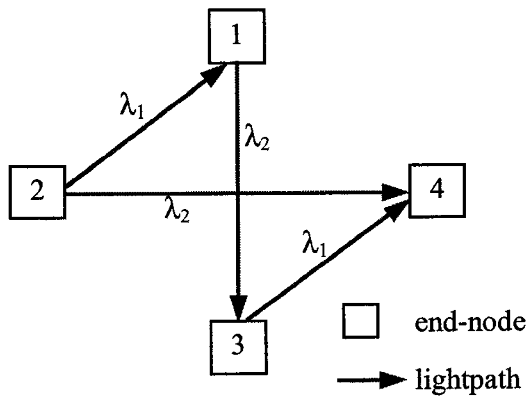


Figure 2.4 Wavelength Routed Networks (Logical Topology)

2.4 Single-hop and Multi-hop Networks

In single-hop networks [Muk92a], the nodes must communicate with one another in one hop (by one lightpath). That means, a packet transmitted from source reaches its destination directly without going through other end nodes. These networks are all-optical, which results very fast communication.

All-optical networks [Muk92a] are designed to avoid the electronics bottleneck (the performance is limited by the maximum speed of electronics) employed in switches and end-nodes. That is, information is conveyed in optical domain (without facing any electro-optical conversions) through the network until it reaches its final destination. In all-optical networks, the end-to-end connection contains only optical devices. Therefore, the electronics at a node only handle the data intended for that node, not the data passed through that node on to other nodes in the network.

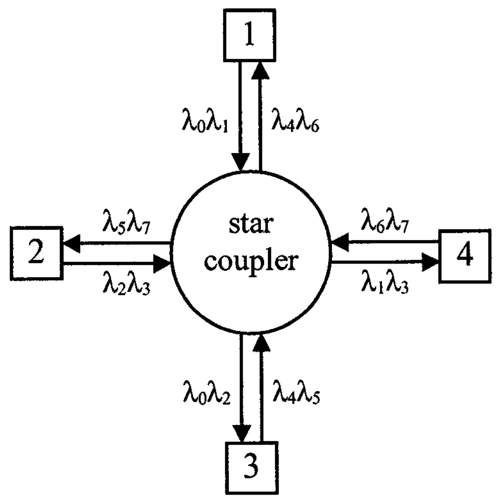
Single-hop networks with N nodes must provide $N(N\text{-}1)$ lightpaths. Therefore, they need a large number of wavelengths for static networks. In a reconfigurable single-hop network, a significant amount of dynamic coordination between nodes is required. For a packet transmission to occur, one of the transmitters of the sending node and one of the receivers of the destination node must be tuned to the same wavelength for the duration of the packet's transmission.

Due to the problems of single-hop networks, an alternative mechanism is needed. In multi-hop networks [Muk92b], a packet from a source to a destination may have to hop through intermediate nodes. The channel to be tuned is relatively static by a node's transmitter or receiver. The intermediate nodes are responsible for routing data among wavelength channels electronically. In this way, the packet is transmitted from optical domain to electronic domain and then to optical domain. Therefore, electronic buffering slows the network.

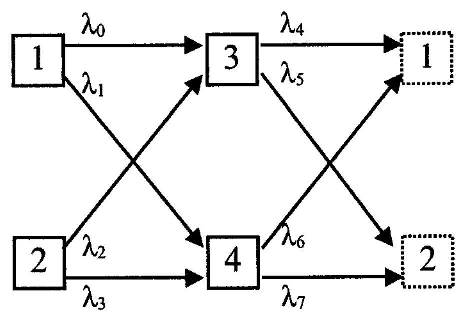
Figure 2.5 is an example of physical and logical topology of a multi-hop network. The physical topology is a star while the embedded logical topology is a 2×2 torus. Note that, in this example, node 1 can communicate with node 3 and node 4 directly via wavelength channels λ_0 and λ_1, but in order to reach node 2, information from node 1 should multi-hop either through node 3 or node 4.

Wavelength routed networks is a combination of single-hop and multi-hop systems. A lightpath provides single-hop communication. Because of limited number of wavelengths, multi-hopping between lightpaths for some source-destination node pairs may be necessary.

In Figure 2.4, communications between $2 \rightarrow 1$, $2 \rightarrow 4$, $1 \rightarrow 3$, $3 \rightarrow 4$ are examples of single-hop since each of them is done by single lightpath. However, transmission between $2 \rightarrow 3$ is multi-hop, because it needs two lightpaths $2 \rightarrow 1$ and $1 \rightarrow 3$ to establish this communication. A lightpath may carries not only the direct traffic between the nodes it connects but also traffic between nodes that are not directly connected in the logical topology.



Physical Topology



Logical Topology

Figure 2.5 A Multi-Hop Network

2.5 Routing and Wavelength Assignment

The logical topology of wavelength routed networks consists of a set of lightpaths, which are set up between end nodes over the physical topology. With wavelength continuity constraint, each lightpath is assigned a path through the network and a wavelength on that path. Finding routes for the lightpaths through the physical topology and assigning wavelengths to these lightpaths is called routing and wavelength assignment problem (RWA) [RS95, ZJM00, OB01, SB97, Ram95].

Depending on whether the lightpath requests are known initially and fixed over time or not, routing and wavelength assignment schemes can be classified into two categories as static and dynamic.

In static RWA [Mar93, Chl93, CGLM+00], all the lightpath requests between the end-node pairs are known initially. This is the case for large transport networks or wide-area networks (WANs), in which the traffic demand is more fixed over time. Therefore, static RWA can be seen as the primary case for today's wavelength routed networks, which are suitable for WANs due to their high costs.

In dynamic RWA [Ger96a, Ger96b], lightpath requests between end-nodes assumed to arrive at random times and have random holding times. That is, lightpath requests are established on demand. This is the case for data networks or local-area networks (LANs), in which the traffic demand frequently changes over time.

2.6 Fault Management Techniques

The routing problem discussed above assumes that the network has no faults. However, network fault is a major problem in optical networks. In a wavelength routed network, the failure of a network component such as a fiber link can lead to the failure of all the lightpaths that traverse the failed link, thereby leading to large data loss. So, it makes fault-tolerance a very important issue in network management [LTS01, MS98, MT01, SA00, SSS02].

There are essentially two types of fault-management techniques: protection and restoration. In protection [RM99b, CB98, EHS00, SG00, ABJ02, SW99], spare capacity is reserved in advance, which means the backup resources (routes and wavelengths) are pre-computed and reserved in advance.

In restoration [RM99a, IG00, LDS98], the spare capacity available after the fault's occurrence is utilized for rerouting the disrupted traffic. That means, when a failure occurs, another route and a free wavelength have to be discovered dynamically for each interrupted connection.

Generally, restoration schemes are more efficient in utilizing capacity due to the sharing of the spare capacity, while protection schemes have a faster recovery time and provide guarantees on the recovery. We consider protection schemes in WDM layer in our research.

2.6.1 WDM Protection

From the network-topology perspective, WDM protection can be classified as ring protection [GRS97, RL01] and mesh protection [EHS00, RM99b, MPPM02]. Both ring protection and mesh protection can be further divided into two groups: path protection [RM99b] and link protection [SB99], which are illustrated in Figure 2.6.

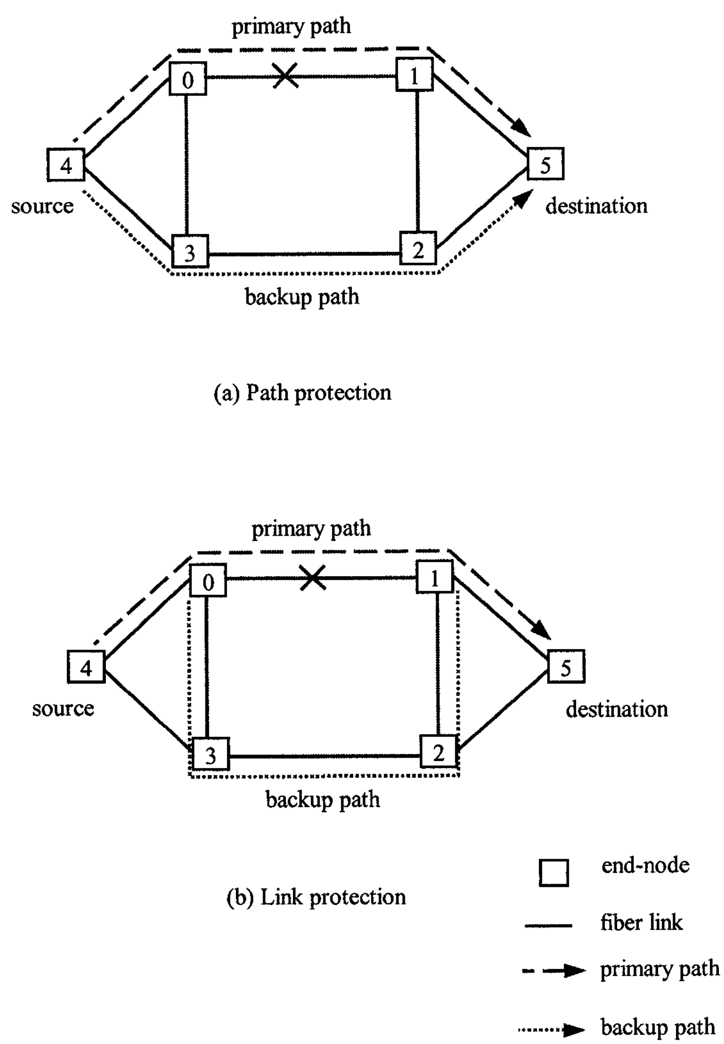


Figure 2.6 WDM Protection

In path protection, the traffic is rerouted through a link disjoint backup route once a link failure occurs on its working path. In link protection, the traffic is rerouted only around the failed link. WDM path protection typically outperforms link protection. Therefore, we use path protection on a mesh topology in our study.

In the design of optical networks, a standard assumption is that there is at most one fiber fault in the physical network at any time. We also make the same assumption in our research.

In WDM path protection, the lightpath that carries traffic during normal operation is known as the primary lightpath. When a primary lightpath fails, the traffic is rerouted over a new lightpath known as the backup lightpath. When setting up the primary lightpath, a backup lightpath is set up as well. The primary path and the corresponding backup path should be fiber-disjoint. Otherwise, if the joint fiber fails, the primary path and the backup path are all fail.

There are two approaches in WDM path protection, one is dedicated-path protection, another is shared-path protection (also called backup multiplexing). In dedicated-path protection [SW99], for each primary path, a dedicated fiber-disjoint backup path and wavelength are reserved. The backup wavelength is not shared with other backup paths.

In shared-path protection [SRM02, YJ01], for each primary path, a fiber-disjoint backup path and wavelength are reserved. However, the backup wavelength may be shared with other backup paths. If two primary paths are fiber-disjoint, then, under the single fiber fault assumption, they will never need to use their backup paths at the same time. Thus, their backup paths can share one or more fibers and be assigned the same wavelength. In Figure 2.7, primary paths p_1 and p_2 are fiber-disjoint, so the corresponding backup paths b_1 and b_2 can share channel λ_2 and edge $0 \rightarrow 2$.

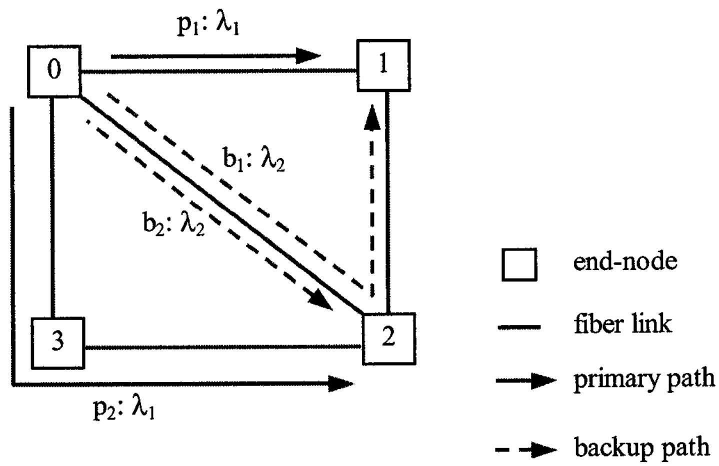


Figure 2.7 Shared-Path Protection

Shared-path protection is more capacity efficient than dedicated-path protection, since the resources used to create backup paths remain idle under fault-free conditions. However, the recovery operation for shared-path protection can be complex and requires reconfigurable optical router at the intermediate routing nodes. Dedicated-path protection schemes are simpler and faster and do not require reconfigurable optical routers. In our research, we try to implement backup multiplexing.

Chapter 3: Overview of Our Work

In this chapter, we present an overview of our research work. First, we define the WDM network model used in our research, which includes the physical topology, the fiber characteristics, and the traffic matrix. Based on this network model, we explain the objectives of our research and the two approaches used in our research. The first approach involves a Mixed Integer Linear Programming (MILP) formulation, which generates the optimum logical topology and the optimum traffic routing scheme. Because solving this MILP formulation takes a very long time, we introduce the second approach, based on a heuristic algorithm, which creates the suboptimum logical topology, but can be solved much faster than the MILP formulation. The first approach has been described in detail in [Gua03]. This thesis concentrates on the second approach.

3.1 WDM Network Model Definition

The model of WDM network that we used in our design method is defined by a physical topology. It is a mesh topology composed of N end nodes, a number of router nodes and M WDM links connecting two nodes (end nodes or router nodes). Each WDM link between nodes i and j is bidirectional and consists of two unidirectional fibers. Each fiber can simultaneously carry the same set of wavelengths, λ_1, λ_2, ..., λ_K. In other words, there are K channels with the channel number m having a carrier wavelength λ_m, $1 \leq m \leq K$. The capacity of each WDM channel (i.e. the maximum amount of traffic it can carry) is the same. In our model, the i^{th} end node has T_i^m transmitters which are tuned to wavelength λ_m and R_i^m receivers which are tuned to wavelength λ_m, $1 \le m \le K$. A typical network is shown in Figure 3.1.

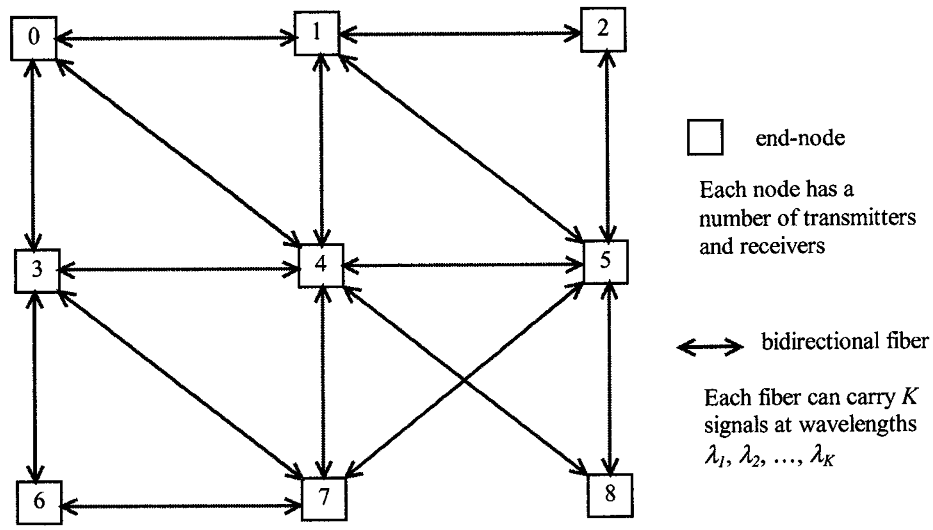


Figure 3.1 A Typical Network

In addition to the physical topology, the traffic matrix T, a $N \times N$ matrix is given, each entry of the matrix t_{ij} is the amount of traffic from node i to node j. As mentioned in chapter 1, we only consider static traffic in this study. Under a static traffic pattern, the connection requests are all available in advance.

3.2 Objective

Based on the physical topology and the traffic matrix T, the goal of this study is to determine the lightpaths to handle all the traffic requirements as specified by the traffic matrix T, and hence define the logical topology. For a given entry t_{ij} in the traffic matrix T, our approach is to find a mechanism for communicating the traffic t_{ij} from node i to node j for all (i, j) pairs. To do this, we have to find, for all i and j where t_{ij} is not 0, a number of routes $r_1, r_2, \ldots, r_p$, for some p, p > 0, from node i to node j. The traffic t_{ij} will be communicated with some of the traffic t_{ij} using route r_1, some using route r_2, ..., the rest using route r_p. In some cases, a route consists of a single hop, so that it is possible to have a primary lightpath and a backup lightpath from node i to node j. In other cases, we need to find nodes $A, B, \ldots, X$, such that we have a lightpath from i to A, a lightpath from A to B, ..., a lightpath from X to j, in order to define a route $i \rightarrow A \rightarrow B \rightarrow \ldots, \rightarrow X \rightarrow j$ from i to j. In this case, we need to have a primary lightpath and a backup lightpath between the node pairs i and A, A and B, ..., X and j

We assume the wavelength continuity constraint, so that the channels on all fibers used by a given lightpath have the same wavelength. For each lightpath, say from node i to node j, we need to find a route and assign a wavelength λ_m for it, and we need to ensure that node i has a transmitter and node j has a receiver at wavelength λ_m which is guaranteed to be available when we need to use this lightpath.

3.3 MILP Formulation

The problem we mentioned above can be solved by using Mixed Integer Linear Programming (MILP). Linear Programming (LP) is the most popular optimization technique designed to optimize the usage of limited resources in operation research. LP problems are composed of objective functions and constraints. If any or all of the

LP variables are further restricted to take integer values in the optimal solution, then it is called MILP [IL01].

In our research, we use the MILP formulation to solve the WDM network optimization problem, which try to find the optimum solution for logical topology design and traffic routing scheme. The objective function of the formulation is to minimize the network congestion F_{max}, which is the maximum amount of traffic flow on any lightpath.

For the physical network model defined in section 3.1, the tasks of this problem are as follows:

- Determine which lightpaths should be chosen. The set of potential lightpaths consists of all lightpaths between all $N(N-1)$ node pairs. Note that this formulation allows several lightpaths between a given pair of nodes. For example, if two lightpaths are allowed between any node pair, the set of potential lightpaths has $2 \times N(N-1)$ lightpaths.

- Determine a primary path and a backup path for each such chosen lightpath, along with a wavelength assigned to each of the primary path and the backup path.

- Determine how the traffic between each node pair will be routed from the source to the destination based on the given traffic matrix.

The constraints of the formulation are:

- For a given source-destination pair, the primary path and the backup path must be fiber-disjoint. If the primary path and the backup path share a fiber, and the shared fiber fails, both the primary path and the backup path will be affected at the same time. In order to avoid this, the primary path and the corresponding backup path must be fiber disjoint.

- Each primary path is edge-channel disjoint with every other primary path. Any fiber cannot carry two signals at the same carrier wavelength. Therefore, all lightpaths sharing a common fiber should be assigned different carrier wavelengths. Using the same argument, if two primary paths use the same carrier wavelength, then they cannot use the same fiber on their routes.

- Each primary path is edge-channel disjoint with respect to every backup path. This ensures that no backup path use the same channel and fiber used by any primary path. The rationale for this is the same as that for the previous constraint.

- If a fiber is shared by two primary paths and this fiber fails, then both these primary paths fail at the same time and the traffic must be switched over to their respective backup paths. However, if their backup paths share a channel and an edge, then both the backup paths cannot be used at the same time. Therefore, if two backup paths share a channel and an edge, then corresponding primary paths must be fiber-disjoint.

- If no fiber fails, the number of primary paths which use wavelength λ_m and originate from node u, should be less than the number of transmitters at node u tuned to λ_m, for all u and m. This ensures that we have the requisite number of transmitters at each node.

- If no fiber fails, the number of primary paths which use wavelength λ_m and terminate at node u should be less than the number of receivers at node u tuned to λ_m, for all u and m. This ensures that we have the requisite number of receivers at each node.

- If a fiber fails, the total number of lightpaths, including the working primary paths and the backup paths that replace the unusable primary paths, originating from node u and using wavelength λ_m cannot exceed the number of transmitters at node u tuned to λ_m, for all u and m.

- If a fiber fails, the total number of lightpaths, including the working primary paths and the backup lightpaths that replace the unused primary paths, terminating at node u using wavelength λ_m cannot exceed the number of receivers at node u tuned to λ_m, for all u and m.

- The amount of traffic t_{ij} from i to j may be handled using a number of routes. If more than one route is used to handle t_{ij}, then each route carries only a part of t_{ij}, and the sum of the traffic carried by these routes is t_{ij}.

- The maximum amount of traffic flow on any lightpath is $F_{max.}$
- The total amount of traffic flow on any lightpath is not greater than 1. In other words, the maximum capacity of a lightpath is 1.

The complete mathematical formulations are in Appendix A, which has been studied in detail in Ms. Hong Guan's thesis [Gua03]. We use CPLEX 7.5 [IL01], a powerful optimization tool, to solve this MILP formulation. The solution of this problem provides us the optimal logical topology and gives us a routing scheme over the logical topology. Our primary objective is to minimize the network congestion.

However, the main drawback of this method is that it takes a very long time to solve the MILP problem. Even for a four node network with 10 edges and 4 channels per fiber, the number of constraints of this complete MILP formulation is 15476. The number of variables is 3840. And the number of integer variables is 348. It needs 467484.68 seconds to get the solution. Therefore, this approach is not practical even for small networks.

3.4 A Heuristic for Defining the Logical Topology

As solving the complete MILP formulation is time consuming, we divided this problem into two parts. First, we develop a heuristic algorithm to establish the logical topology with WDM shared-path protection. This is the major work of this thesis. After that, we formulate the traffic routing problem as a LP problem over the suboptimum logical topology, which can be solved much faster than the MILP problem. This work will be done by another graduate student.

In this thesis, one assumption we make, for simplicity, is that the entire traffic t_{sd} from node s to node d is handled using one route only.

In our heuristic algorithm, we use a greedy heuristic and try to find routes for the node pairs having high traffic first. Since there is relatively more spare capacity at the beginning, it is easier to find single-hop routes for the node pairs having high traffic, which helps to reduce the congestion in the network. In general however the traffic between some node pairs involve multi-hop paths. If we use a multi-hop path $s \rightarrow A \rightarrow B \rightarrow \ldots \rightarrow X \rightarrow d$ to route the traffic t_{sd}, zero or more of the hops use some or all of the spare capacity of existing lightpaths and the remaining hops use new lightpaths created during our process of finding a near optimal way to handle t_{sd}. The cost of the path to handle t_{sd} is the number of new lightpaths created during our process to handle t_{sd}.

In this algorithm, we try to find the minimum cost lightpaths for each source-destination pair (s, d). For example, let our heuristic determine that the traffic t_{sd} should use multi-hop route $s \rightarrow a \rightarrow b \rightarrow d$. If $s \rightarrow a$ and $b \rightarrow d$ are the new lightpaths and $a \rightarrow b$ is an existing lightpath, then the total cost to transmit traffic t_{sd} is $(1 + 0 + 1) = 2$.

Given the physical network and the traffic matrix, our heuristic includes the following steps:

Step1: Create, if possible, n edge-disjoint shortest paths for all the $N(N-1)$ node pairs using Dijkstra's algorithm (in our implementation we used $n = 3$).

Step2: Sort all the traffic from the highest to the lowest.

Step3: While all nonzero entities in T have not been considered, repeat steps 4 – 7.

Step4: Take the highest traffic t_{sd} in T, t_{sd} is the amount of traffic from node s to node d.

Step5: Check if it is possible to send traffic t_{sd} from node s to node d using existing lightpaths. If so, omit step 6.

Step6: Check if it is possible to send traffic t_{sd} from node s to node d using a minimum number of additional lightpaths.

Step7: If step 6 failed, report failure and stop. Otherwise set t_{sd} to zero.

Some explanation of these steps is given below. In step 1, these shortest paths provide the routes for the primary and backup lightpaths between source-destination pairs. Because each primary path and its backup path between a source-destination pair must be edge-disjoint, it is convenient to consider edge-disjoint paths only. Three shortest paths ($n = 3$) for every node pair provide enough route choices for the primary path and the backup path in most networks [RM98].

In step 5, we check if there is enough spare capacity in some existing lightpaths to carry the traffic t_{sd}. If there exist nodes A, B, ... X such that there are existing lightpaths $s \rightarrow A, A \rightarrow B, \ldots, X \rightarrow d$, and each of the lightpaths have at least t_{sd} spare capacity then these existing lightpaths can carry the additional traffic t_{sd} without adding any additional lightpath to the network. We choose the sequence lightpaths $s \rightarrow A \rightarrow B \rightarrow \ldots \rightarrow X \rightarrow d$, which can transmit the traffic t_{sd} by a minimum number of hops.

In step 6, if traffic t_{sd} cannot be transmitted by existing lightpaths, we need to find a minimum number of new lightpaths for it. In other words, we try to decrease as much as possible the number of hops to transmit the traffic t_{sd} from node s to node d. We wish to use as few as hops since each hop involves one additional transmitter and receiver. Each new hop has a primary lightpath and a backup lightpath. For any two edge-disjoint primary paths, we try to find shared backup paths for them.

The main steps of this algorithm are explained in detail in chapter 4. We tested the heuristic with three networks, a random six node network, the NSFNET (14 nodes) and the EURONET (20 nodes). The details of the experiments are presented in chapter 5.

Chapter 4: Heuristic Implementation

In this chapter, we describe our heuristic in detail. The goal of this heuristic is to find the most economic way to define a logical topology for a given WDM network and traffic matrix. We use a heuristic that looks at the source-destination pair having the highest traffic first. In this process, we first try to use the existing lightpaths to send the traffic for this source-destination pair. If this is not possible, we add the minimum number of new lightpaths which can accommodate the traffic for this source-destination pair. In the remainder of this chapter, we will first introduce the simplifications which are used in our heuristic, and then illustrate each main step of the implementation in detail.

4.1 Simplifications Used in the Heuristic

We have made three simplifying assumptions in our heuristic as follows:

Simplification 1) In the MILP formulation, we consider all the traffic requests simultaneously. In our heuristic, we only consider one traffic request at a time, starting with the node pair having the highest traffic. This makes our heuristic a greedy heuristic.

Simplification 2) In our heuristic, we only choose the route for each lightpath from three edge-disjoint alternative paths between each source-destination pair.

However in the MILP formulation, all possible alternative paths between each source-destination pair are considered.

Simplification 3) In the heuristic, we send the entire traffic t_{sd} from node s to node d by one route only. That is we do not decompose the traffic t_{sd} to more than one route. In the MILP formulation, the traffic t_{sd} can be distributed using more than one route.

4.2 Create Shortest Paths

The first step in our algorithm creates three edge-disjoint shortest paths for every node pair in the network. When we want to establish a lightpath between a particular source-destination pair, we choose one of these paths as the route for the primary lightpath and another as the route for the backup lightpath. Since the primary route and the backup route must be edge-disjoint, it is convenient to consider only edge-disjoint paths. We accomplish this by successively applying Dijkstra's algorithm [Ski97] to each node pair. When attempting to find a new shortest path for a given node pair, we erase the edges used by the previously found shortest paths for this node pair and use the modified graph to create the next shortest path. This ensures that all three shortest paths will be edge-disjoint. A brief description of Dijkstra's algorithm and an example of creating three shortest paths is given in Appendix B.

4.3 Search Best Path

After creating three shortest paths for every node pair, we sort the traffic in descending order. Then we try to find the best path for each traffic request. This is the main part of our algorithm. We select the source-destination pair ($s \rightarrow d$) having the highest traffic t_{sd}. Our objective is to find a single-hop or a multi-hop route from source s to destination d traversing one or more logical edges using a minimum number of new lightpaths. In general, such a route will consist of a combination of new lightpaths and existing lightpaths.

First, we try to route the traffic t_{sd} from node s to node d using existing lightpaths which have already been established to accommodate traffic requests considered previously. As explained earlier, to implement the path protection scheme, each edge in the logical topology corresponds to two lightpaths – the primary lightpath and the backup lightpath. It is convenient, from now on, to speak of the load on a logical edge. This is the traffic that will be carried by the primary lightpath corresponding to this logical edge if all fibers in the path are fault-free. If there is any fault involving this primary path, the same traffic will be carried by the backup lightpath. In order to accommodate the traffic t_{sd} from source node s to destination node d using the spare capacity of existing logical edges, we only need to look at logical edges having spare capacity greater than or equal to t_{sd}. If the destination d cannot be reached using only these existing logical edges, we try to create some new logical edges. Each new logical edge that we consider is a potential candidate that will be included in the final logical topology only if that logical edge appears in a minimum cost path to the destination node d. Figure 4.1 shows our heuristic to find the best route from source node s to destination node d carrying the traffic t_{sd}.

Step1: Create a zero-cost list containing only s, and create an empty one-cost list.

Step2: Create a zero-cost list containing all nodes that can be reached from current zero-cost list using existing lightpaths only. If d is in the zero-cost list, then go to step 9.

Step3: For each node i in zero-cost list, do steps 4 – 5.

Step4: For each node j that does not appear in the zero-cost list, do step 5.

Step5: If it is possible to add a new logical edge from i to j, then add j to the one-cost list.

Step6: If d is in the one-cost list, then go to step 9.

Step7: If the one-cost list is empty, then the algorithm has failed.

Otherwise,

redefine zero-cost list so that contains the same nodes that appear in the one-cost list, and

empty one-cost list.

Step8: Repeat steps 2 – 7.

Step9: Set up all new logical edges.

Figure 4.1 Search-Best-Path

Figure 4.2 is an example to illustrate how to find a route for a traffic request $t_{0\text{-}10}$ from node 0 to node 10. The set of logical edges currently existing in the network is {L_0: 0 → 1, L_1: 0 → 2, L_2: 2 → 3, L_3: 4 → 6, L_4: 6 → 8, L_5: 5 → 7}. At the beginning, the zero-cost list contains only source node 0 (Figure 4.2(a)), and one-cost list is empty. Suppose that the logical edges L_0, L_1 and L_2 have sufficient spare capacity to carry the traffic $t_{0\text{-}10}$ from node 0 to node 10. After step 2, the zero-cost list contains nodes {0, 1, 2, 3}, which can be reached by using L_0, L_1 and L_2. Figure 4.2(b) shows this situation where we have used dashed lines to depict an existing logical edge. Because destination node 10 is not in the zero-cost list, we go to step 4. Suppose we are able to establish the following logical edges: L_6: 0 → 4, L_7: 1 → 5. Figure 4.2(c) shows this situation where we have used solid lines to depict new logical edges. After step 4, the one-cost list contains {4, 5}. Because destination node 10 is not in the one-cost list, we move the nodes in the one-cost list to the zero-cost list and empty the one-cost list. Now the zero-cost list becomes {4, 5} and the one-cost list is empty. Since we still haven't reached the destination node 10, we need to repeat from step 2. After step 2, nodes {6, 7, 8} are added to the zero-cost list by using existing logical edges L_3, L_4 and L_5 (Figure 4.2(d)). Now the zero-cost list becomes {4, 5, 6, 7, 8}. Since the destination node 10 is not in the zero-cost list, we have to do step 4. Suppose we are able to establish the following logical edges: L_8: 5 → 9, L_9: 7 → 10 (Figure 4.2(e)). After step 3, the one-cost list contains {9, 10}. Because the destination node 10 is in the one-cost list, we stop searching and set up the new logical edges required to reach the destination node 10. The traffic from node 0 to node 10 goes through four logical edges: 0 → 1 (zero cost), 1 → 5 (one cost), 5 → 7 (zero cost), 7 → 10 (one cost). The new logical edges we need are 1 → 5, 7 → 10.

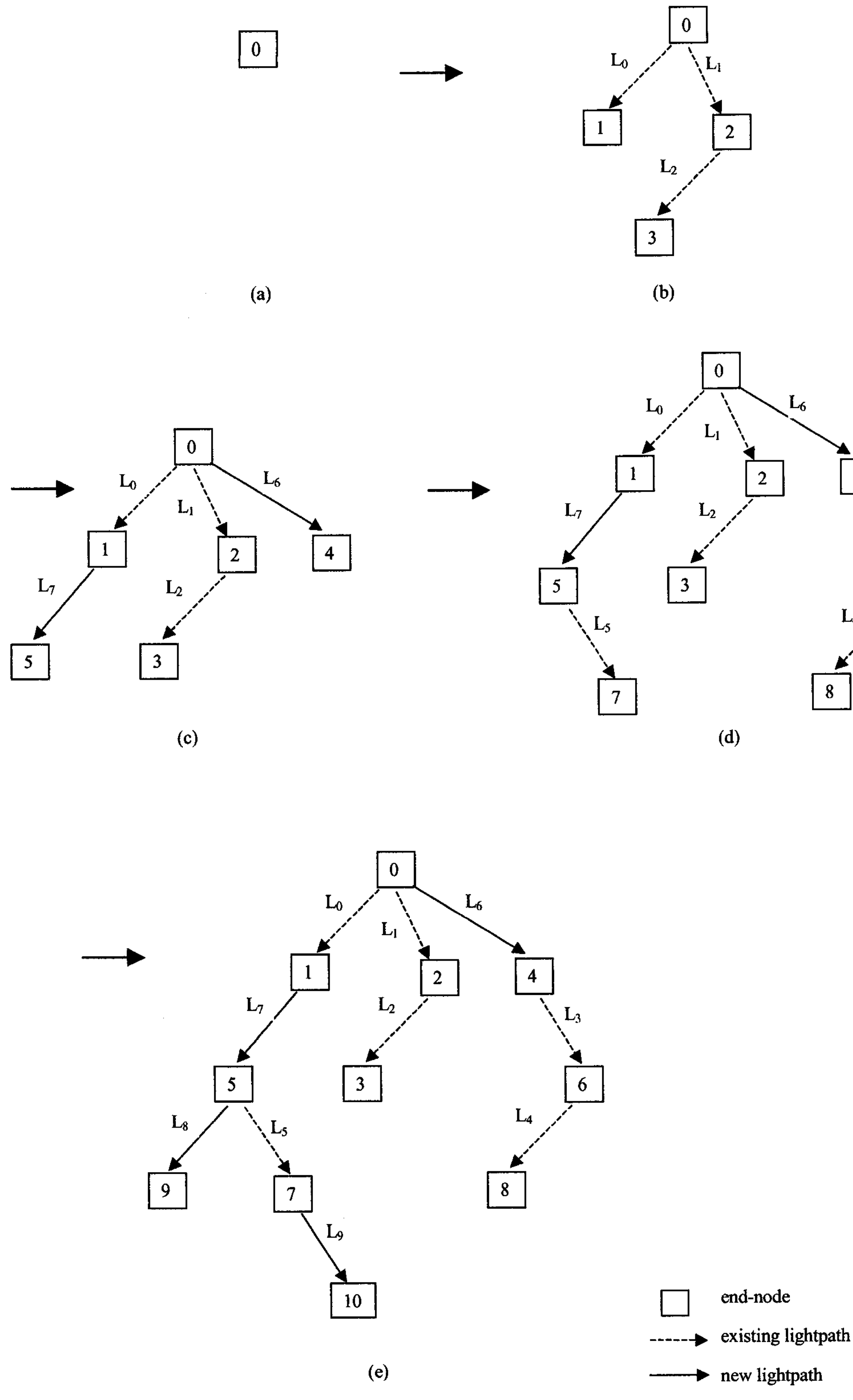


Figure 4.2 An Example for the Search-Best-Path Algorithm

The heuristic given in Figure 4.1, has two main steps. The first is to extend the search tree using the existing logical edges that can carry the additional traffic t_{sd}. The other is to find the set of new logical edges that may be constructed from the nodes already in the search tree. We will now explain these steps in detail.

4.4 Reuse of Existing Logical Edges

One of the tasks in the search-best-path algorithm (step 2) is to find all new nodes that can be reached from a given set of nodes, using only existing logical edges. In this section, we explain this process in detail. First we check all the existing logical edges, whose sources are the nodes in the zero-cost list. From these, we select the logical edges that can carry an additional t_{sd} amount of traffic. The destinations of these logical edges are added to zero-cost list. This process is then repeated with the new set of nodes, until no new nodes can be added. Figure 4.3 gives the outlines of this work.

Step1:	Create a current-node list consisting of the initial set of nodes in zero-cost list
Step2:	Create an empty next-node list
Step3:	For each node *i* in the current-node list, do steps 4 – 5
Step4:	For each existing logical edge *lp* from node *i*, do step 5
Step5:	If the spare capacity of this logical edge is greater than the traffic t_{sd} and the destination of *lp* is a node that is neither in the zero-cost list nor in the next-node list, then put the destination of *lp* to next-node list

Step6:	If next-node list is empty, stop. Otherwise,
	    zero-cost list = zero-cost list + next-node list
	    current-node list = next-node list
	    go to step 2

Figure 4.3 Reuse of Existing Logical Edges

4.5 Create New Logical Edges – Phase I

If it is not possible to find a path $s \rightarrow A \rightarrow B \rightarrow \ldots X \rightarrow d$ from source node s and reach the destination node d using the spare capacities on existing logical edges, we need to check if we can create new logical edges to find a path $s \rightarrow A \rightarrow B \rightarrow \ldots X \rightarrow d$ in the logical topology to accommodate the traffic t_{sd}. Some of the edges in this path are logical edges which existed earlier and have sufficient spare capacities while the remaining edges are new. Our objective is to create new logical edges to define the path using as few network resources as possible. This corresponds to steps 3 – 5 in Figure 4.1. However the computational effort to find the best solution to the problem is enormous. This is the most complicated part of our heuristic and is best described in two phases. In phase I, we find a path $s \rightarrow A \rightarrow B \rightarrow \ldots X \rightarrow d$ where we ignore the possibility of sharing backup paths.

4.5.1 Initialize Wavelength Information on Fibers

In order to find a new logical edge, we need the wavelength information on all fiber links, i.e., which wavelengths are available, which wavelengths are being used and whether they are used by primary paths or backup paths. The wavelength information for the existing logical edges is readily available. Since we may have to explore a number of possible multi-hop routes from the source to the destination before we find the minimum cost multi-hop routes, we have to take into account the new logical edges from the source node *s* to the current node *i*. For example, in Figure 4.2, when we try to find new logical edges from node 7, in addition to all the existing logical edges (shown by dashed lines), we also need to know the wavelength information for the new logical edge L_7: $1 \rightarrow 5$. Information corresponding to other new logical edges (e.g. L_6: $0 \rightarrow 4$, L_8: $5 \rightarrow 9$) are not relevant at this point, since they are not on the path $0 \rightarrow 1 \rightarrow 5 \rightarrow 7$. This means that, they will never be actually created if the traffic is ultimately sent over the sequence of logical edges $0 \rightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 10$.

Figure 4.4 outlines how to find a new logical edge from node *i* in the zero-cost list to a new node *j* without considering the possibility of shared backup paths.

Step1:	Create temporary wavelength information on all fiber links for all the existing logical edges and the relevant new logical edges
Step2:	For each new node j, do step 3
Step3:	If we can find a new primary lightpath and a new backup lightpath from node i to node j add node j to one-cost list

Figure 4.4 Create New Logical Edges – Phase I

4.5.2 Find a New Logical Edge

As mentioned before, whenever we try to establish a logical edge between a given node pair (i, j), we are actually trying to find two lightpaths (a primary path and a backup path) for this source-destination pair.

To establish a lightpath from a source node i to a destination node j, we have to

- find a route for the lightpath
- make sure that there is a wavelength λ_m that can be used for the lightpath
- make sure that there is an available transmitter at node i tuned to λ_m
- make sure that there is an available receiver at node j tuned to λ_m.

If two such lightpaths are found (along two edge-disjoint routes) then it is possible to establish a new logical edge from node i to node j. The outline of this process is shown in Figure 4.5. When the process stops, either the process has failed or it has succeeded. If it has succeeded, one of the paths found may be used as a primary path and one of the remaining paths may be used as a backup path.

Step1: If there is no transmitters at node i or there is no receivers at node j, then indicate failure and stop

Step2: For each shortest path p from i to j, do step 3

Step3: If the set of available wavelengths (Λ) on path p is not empty, repeat step 4

Step4: If there is an available transmitter for λ_m at node s and an available receiver for λ_m at node d, for some $\lambda_m \in \Lambda$, then

increase the number of paths found by 1

Step5: If the number of paths is at least two, then

indicate success and stop

Otherwise, indicate failure and stop

Figure 4.5 Find a New Logical Edge

4.6 Create New Logical Edges – Phase II

In phase I we found a path $s \rightarrow A \rightarrow B \rightarrow \ldots X \rightarrow d$ from source node s and reach the destination node d using a minimum number of new logical edges. For each new logical edge, we create the backup path from scratch and hence we are using dedicated protection. It is possible to further minimize the needed network resources by using shared protection. In this phase, we use the same primary paths that we found in phase I but check whether shared protection is possible for the backup path. Figure 4.6 shows how to set up the lightpaths on the route from node s to node d.

Step1:	For each new logical edge on the route $s \rightarrow A \rightarrow B \rightarrow \ldots X \rightarrow d$ from s to d, do step 2
Step2:	Create the primary lightpath for the new logical edge using one of the feasible routes and the corresponding wavelength found in phase I
Step3:	For each new logical edge on the route from s to d, do step 4
Step4:	Create the backup lightpath with shared protection using the remaining feasible routes found in phase I. We minimize the amount of additional network resources used by ensuring that the number of channels not shared with other backup paths is minimum.

Figure 4.6 Create New Logical Edges – Phase II

Chapter 5: Experiments and Results

In this chapter, we have explained the experiments carried out for testing our heuristic and have analyzed the results of these experiments. In our experiments, we used three physical topologies. One is a randomly generated six node network. The other two are well-known networks – the NSFNET (14 node) [RS96] and the EURONET (20 node) [MSYZ95].

We have used our heuristics to generate logical topologies, which are embedded over a given physical network. We are interested in studying the performances of the logical topologies under different conditions. These conditions include:

- the number of available wavelengths on each fiber,
- the number of transmitters at each node for each wavelength and
- the load (total traffic) on the network

We always ensured that there were enough receivers at each node for each wavelength.

To test the heuristics described in chapters 3 and 4, we have written a C program which takes, as inputs, the physical topology of the network, the number of channels per fiber, the number of transmitters and receivers per wavelength per node, and the traffic matrix. When evaluating our networks, we are interested in the following information:

- how much total traffic can be supported by the logical topology
- how many traffic requests are failing, because they cannot be accommodated
- how efficiently the network resources are being utilized
- how long it takes to obtain a solution.

We also need to determine how the above values change with the amount of available resources. In order to study the networks generated by our heuristic, we have run a number of experiments

For each set of experiments, we selected a specific physical topology and varied the traffic matrix as well as the number of available wavelengths and available transmitters at each node. For each physical topology, the traffic matrices were selected to reflect different load conditions. They were classified as low, medium or high load, based on the total amount of traffic presented to the network (obtained by adding all the entries of the traffic matrix).

Our program generated the logical topology, as defined by the primary and backup lightpaths. The results of the output generated by the program are summarized below in Tables 5.1 – 5.3. These tables report on a number of different parameters, including the number of single-hop connections, the number of multi-hop connections, the number of lightpaths, and the total amount of traffic accommodated by the resulting logical topology. They also indicate the number of failed requests, and the time required to obtain a solution (indicated in the last column). This is an important criterion, since our main objective in attempting this work was to be able to generate a reasonable solution quickly and efficiently.

One way to evaluate how efficiently resources are being utilized, is to determine the percentage of idle capacity in the network. Low values of idle capacity indicate more efficient resource utilization. The information on resource utilization is analyzed in section 5.4.

5.1 Experiments with a Six Node Network

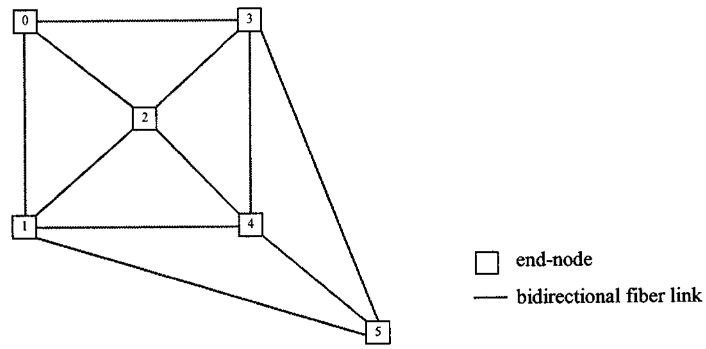


Figure 5.1 Six Node Network

Figure 5.1 is a randomly generated six node network with twenty-two edges. We tested our heuristic on this topology with seven different traffic matrices $\{T_1, T_2, T_3, T_4, T_5, T_6, T_7\}$, where T_1 and T_2, represent low load conditions, T_3 and T_4 represent medium load conditions and T_5 and T_6, represent high load conditions. T_7 represents a special case, where each entry of the traffic matrix, t_{ij} is equal to 100, $i \neq j$. The actual traffic matrices are given in Appendix C. The results for the six node network are given in Table 5.1. For each traffic matrix, we started by allowing 3 wavelengths per fiber and 1 transmitter per wavelength, per node. Then, we gradually increased the resources until the resulting network was able to accommodate the entire traffic. As indicated in Table 5.1, there was not much variation in the amount of time required to find a solution, for the six node network.

Table 5.1 Experiments with a six node network

Traffic matrix	# of wave-length/fiber	# of trans-mitters/node	# of 1-hop connections	# of Multi-hop connections	# of lightpaths	Total traffic handled	# of failures	Solution time (sec)
T_1	3	1	6	3	6	490	21	0.01
T_1	3	2	15	15	16	820	0	0.01
T_1	5	1	12	12	12	760	6	0.01
T_2	3	1	6	1	6	510	23	0.01
T_2	3	2	16	11	17	975	3	0.01
T_2	3	3	16	14	17	1015	0	0.01
T_2	5	1	11	6	12	835	13	0.01
T_2	5	2	20	10	20	1015	0	0.01
T_3	3	1	6	2	6	480	22	0.01
T_3	3	2	15	14	17	1210	1	0.01
T_3	3	3	18	12	19	1220	0	0.01
T_3	5	1	12	8	12	940	10	0.01
T_3	5	2	20	10	20	1220	0	0.01
T_4	3	1	6	1	6	510	23	0.01
T_4	3	2	17	6	17	1260	7	0.01
T_4	3	3	18	5	18	1310	7	0.01
T_4	5	1	12	3	12	940	15	0.01
T_4	5	2	23	7	23	1520	0	0.01

Table 5.1: Continue

Traffic matrix	# of wave-length/fiber	# of trans-mitters/node	# of 1-hop connections	# of Multi-hop connections	# of lightpaths	Total traffic handled	# of failures	Solution time (sec)
T_5	3	1	6	1	6	540	23	0.01
T_5	3	2	13	5	16	1220	12	0.01
T_5	3	3	16	6	18	1430	8	0.01
T_5	5	1	12	2	12	1030	16	0.01
T_5	5	2	25	5	25	1920	0	0.01
T_6	3	1	6	0	6	550	24	0.01
T_6	3	2	16	0	16	1385	14	0.01
T_6	3	3	19	1	21	1675	10	0.01
T_6	5	1	12	0	12	1060	18	0.01
T_6	5	2	26	1	28	2155	3	0.01
T_6	5	3	30	0	30	2355	0	0.01
T_7	3	1	6	0	6	600	24	0.01
T_7	3	2	17	0	17	1700	13	0.01
T_7	3	3	17	2	21	1900	11	0.01
T_7	5	1	12	0	12	1200	18	0.01
T_7	5	2	22	3	28	2500	5	0.01
T_7	5	3	30	0	30	3000	0	0.01

5.2 Experiments with NSFNET

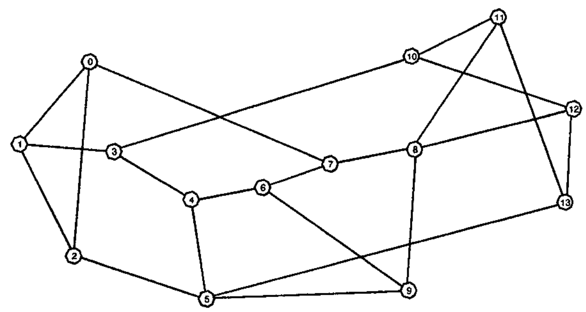


Figure 5.2 Fourteen Node Network

Figure 5.2 is a well-known network NSFNET with fourteen nodes and forty-four edges. Same as the experiments with the six node network, we tested our heuristic on this topology with seven different traffic matrices $\{T_1, T_2, T_3, T_4, T_5, T_6, T_7\}$, where T_1 and T_2, represent low load conditions, T_3 and T_4 represent medium load conditions and T_5 and T_6, represent high load conditions. T_7 represents a special case, where each entry of the traffic matrix, t_{ij} is equal to 100, $i \neq j$. The actual traffic matrices are given in Appendix C. The results for the fourteen node network are given in Table 5.2. For each traffic matrix, we gradually increased the resources (wavelengths and transmitters) until the resulting network was able to accommodate the entire traffic. As indicated in Table 5.2, there was not much variation in the amount of time required to find a solution, for the fourteen node network.

Table 5.2 Experiments with a fourteen node network

Traffic matrix	# of wave-length/fiber	# of trans-mitters/node	# of 1-hop connections	# of Multi-hop connections	# of lightpaths	Total traffic handled	# of failures	Solution time (sec)
T_1	8	1	47	84	51	2015	51	0.26
T_1	8	2	56	119	67	2455	7	0.25
T_1	8	3	53	116	68	2395	13	0.25
T_1	12	1	62	120	67	2525	0	0.26
T_2	8	1	50	39	52	3385	93	0.25
T_2	8	2	64	76	75	4270	42	0.26
T_2	8	3	68	76	76	4295	38	0.26
T_2	12	1	69	72	77	4295	41	0.27
T_2	12	2	88	94	91	4705	0	0.26
T_3	12	1	68	38	78	5260	76	0.27
T_3	12	2	93	66	109	6870	23	0.26
T_3	12	3	106	71	116	7280	5	0.26
T_3	16	1	96	63	106	6815	23	0.27
T_3	16	2	115	67	115	7330	0	0.27
T_4	16	1	97	34	107	695	51	0.27
T_4	16	2	133	47	146	9990	2	0.27
T_4	16	3	138	44	142	10060	0	0.28
T_4	20	1	120	47	133	9430	15	0.28
T_4	20	2	142	40	142	10060	0	0.27

Table 5.2: Continue

Traffic matrix	# of wave-length/fiber	# of trans-mitters/node	# of 1-hop connections	# of Multi-hop connections	# of lightpaths	Total traffic handled	# of failures	Solution time (sec)
T_5	16	1	96	12	109	8105	74	0.26
T_5	16	2	126	18	143	102600	38	0.27
T_5	16	3	129	21	150	10600	32	0.28
T_5	20	1	125	14	134	10000	43	0.28
T_5	20	2	163	18	170	12270	1	0.28
T_5	20	3	165	17	169	12310	0	0.27
T_6	20	1	119	6	132	11070	56	0.28
T_6	20	2	163	7	177	14430	11	0.28
T_6	20	3	169	5	179	14720	7	0.28
T_6	22	1	128	8	144	11950	45	0.28
T_6	22	2	179	2	181	15140	0	0.28
T_7	20	1	122	7	136	12900	53	0.28
T_7	20	2	157	13	186	17000	12	0.29
T_7	20	3	164	9	184	17300	9	0.29
T_7	22	1	125	11	147	13600	46	0.29
T_7	22	2	180	2	184	18200	0	0.29

5.3 Experiments with EURONET

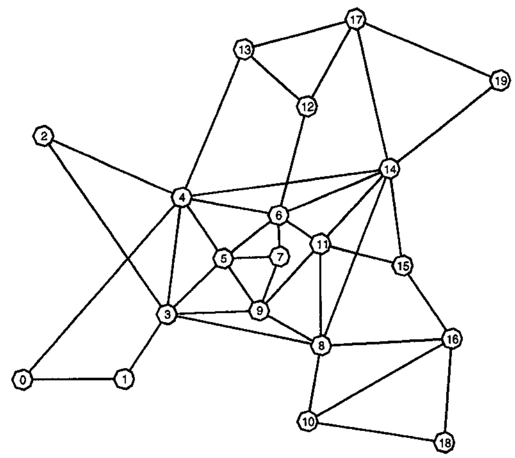


Figure 5.3 Twenty Node Network

Figure 5.3 is a well-known network EURONET with twenty nodes and eighty edges. We tested our heuristic on this topology with ten different traffic matrices $\{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}\}$, where T_1, T_2 and T_3 represent low load conditions, T_4, T_5 and T_6 represent medium load conditions and T_7, T_8, and T_9, represent high load conditions. T_{10} represents a special case, where each entry of the traffic matrix, t_{ij} is equal to 100, $i \neq j$. The actual traffic matrices are given in Appendix C. The results for the twenty node network are given in Table 5.3. For each traffic matrix, we gradually increased the resources (wavelengths and transmitters) until the resulting network was able to accommodate the entire traffic. As indicated in Table 5.3, there was a little bit variation in the amount of time required to find a solution, for the twenty node network.

Table 5.3 Experiments with a twenty node network

Traffic matrix	# of wave-length/fiber	# of trans-mitters/node	# of 1-hop connections	# of Multi-hop connections	# of lightpaths	Total traffic handled	# of failures	Solution time (sec)
T_1	8	1	72	89	79	4110	219	1.71
T_1	8	2	100	220	122	5691	60	2.10
T_1	8	3	104	234	129	5871	42	2.03
T_1	12	1	105	222	119	5761	53	2.22
T_1	12	2	132	248	142	6291	0	2.16
T_2	12	1	102	85	116	6630	192	2.51
T_2	12	2	138	196	171	8500	45	2.66
T_2	12	3	148	217	179	8810	14	3.22
T_2	16	1	139	163	154	8180	77	2.68
T_2	16	2	170	208	180	8980	1	3.32
T_2	16	3	171	208	181	8950	0	3.96
T_3	20	1	169	146	197	11721	65	4.15
T_3	20	2	206	172	215	12451	2	3.82
T_3	20	3	209	169	215	12451	2	4.16
T_3	22	1	190	171	207	12281	19	3.49
T_3	22	2	209	171	216	12471	0	4.15
T_4	20	1	177	116	198	12795	87	2.96
T_4	20	2	221	152	239	14670	7	6.51
T_4	20	3	223	152	243	14770	5	4.74
T_4	22	1	192	128	218	13655	60	3.19
T_4	22	2	230	150	239	14820	0	3.62

Table 5.3: Continue

Traffic matrix	# of wave-length/fiber	# of trans-mitters/node	# of 1-hop connections	# of Multi-hop connections	# of lightpaths	Total traffic handled	# of failures	Solution time (sec)
T_5	22	1	189	97	216	15240	99	3.09
T_5	22	2	248	138	281	18955	2	3.41
T_5	22	3	247	132	276	19000	1	4.02
T_5	24	1	196	106	237	16200	78	3.37
T_5	24	2	257	123	270	19030	0	5.12
T_6	24	1	209	63	238	17815	113	3.57
T_6	24	2	264	103	329	22845	13	6.81
T_6	24	3	271	97	321	22885	12	5.93
T_6	30	1	266	78	290	21775	36	4.35
T_6	30	2	301	79	312	23455	0	8.67
T_7	30	1	322	55	342	26585	3	6.94
T_7	30	2	319	57	345	26515	4	6.74
T_7	30	3	322	55	342	26585	3	6.93
T_7	32	1	279	49	313	24215	52	4.59
T_7	32	2	327	52	344	26680	1	6.93
T_7	32	3	331	49	340	26710	0	7.81
T_8	32	1	282	20	317	26265	77	4.99
T_8	32	2	340	26	373	30280	13	11.16
T_8	32	3	340	29	378	30460	10	12.22
T_8	34	1	370	9	370	31065	0	8.19

Table 5.3: Continue

Traffic matrix	# of wave-length/fiber	# of trans-mitters/node	# of 1-hop connections	# of Multi-hop connections	# of lightpaths	Total traffic handled	# of failures	Solution time (sec)
T_9	32	1	272	19	312	27080	89	4.64
T_9	32	2	392	18	380	32935	20	6.20
T_9	32	3	391	26	393	33525	13	11.44
T_9	34	1	380	0	380	34540	0	8.20
T_{10}	32	1	271	20	311	29100	89	3.94
T_{10}	32	2	321	38	399	35900	21	3.17
T_{10}	32	3	336	25	338	36100	19	4.99
T_{10}	36	1	380	0	380	38000	0	37.47

5.4 Analysis of Experimental Results

Table 5.1 – 5.3 indicate that, when we increase the number of wavelengths per fiber or the number of transmitters per node per fiber, the logical topology created by our heuristic accommodates more traffic. This is because, with more wavelengths or more transmitters, it is possible to establish more lightpaths, which can carry more traffic.

As we increase total amount of traffic, the percentage of idle capacity in the network decreases. The idle capacity C_i is defined as follows:

$$C_i = \left(1 - \frac{\text{total traffic on the network}}{\text{total capacity on established lightpaths}}\right) \times 100 \qquad (1)$$

This means that network resources are being utilized more efficiently, with increasing traffic.

With the high load traffic matrices, the chances of multi-hop communications decrease considerably. This is because, most of the traffic matrix entries are greater than 60 and are routed on one route. So the chances that existing lightpaths will have enough spare capacity to accommodate the traffic are low. Therefore, if enough wavelengths and transmitters are available, the traffic will simply be routed by a single-hop, as this corresponds to a minimum cost.

Chapter 6: Conclusion and Future Work

The objective of this thesis was to develop practical techniques for designing fault tolerant logical topologies. The design process had to take into account the underlying physical topology, the amount and distribution of traffic to be handled by the network and resource constraints on the number of available optical components at each node and the capacity of the optical fibers. Attempts to formalize the design process as a MILP optimization problem can produce precise mathematical formulations. Unfortunately, even with present day high-performance computing capabilities, such formulations are of theoretical interest only. The resulting optimization problems cannot be solved, except for trivial examples.

In this thesis, we have developed a heuristic to design a fault tolerant logical topology and to rout the traffic specification over the logical topology for any given WDM network. We have used WDM shared path protection technique to provide protection against any single fiber failure. This was done, by determining a primary lightpath and a backup lightpath, for each edge in the logical topology.

We have attempted to design our logical topology, using the minimum amount of resources. When processing a specific request, we try to use existing logical edges if possible. If this is not possible, we minimize the number of new logical edges that must be created. That is we use the minimum network resources to handle each traffic specification between a source-destination pair. These resources include the carrier wavelengths on each fiber, the transmitters and the receivers at each node.

We have tested our heuristic on a number of different physical networks. The results show that a feasible solution can be found very quickly (several seconds), even for the fourteen node network and the twenty node network. On the other hand, MILP optimization techniques require several days of computing for just a small four node network.

The results also indicate that we can accommodate the entire traffic, if we allow up to 36 wavelengths per fiber, for the twenty node EURONET. This is quite practical and achievable using present day techniques.

Therefore, we conclude that our heuristic can successfully design fault-tolerant logical topologies for practical-sized networks, using current technologies. Furthermore, it is able to achieve this within a reasonable time frame, making it a viable option for meeting today's logical topology design needs.

In our heuristic, when creating a new logical edge, we try to use as few network resources as possible. However the computational effort to find the best solution to the problem is enormous. Therefore, during the first phase, we find two dedicated lightpaths for the source-destination pair and ignore the possibility of sharing backup paths. The algorithm may be improved by incorporating backup-multiplexing when creating logical edges.

In our implementation, the algorithm cannot handle traffic matrix entries greater than 100. In future, the algorithm should be modified, so that this restriction is removed. In order to do this, we have to allow each traffic matrix entry to be processed over multiple iterations.

The way our heuristic is implemented, each logical topology automatically assumes a routing of the traffic over the resulting topology. However, this may not necessarily be the best possible routing for the topology. In order to improve the performance of the network, an additional step should be carried out to determine an optimal routing (i.e. a routing yielding the lowest congestion) over the logical topology. This is a significant work in itself, and is being carried out by another graduate student.

Appendix A: Complete MILP Formulation

A.1 The problem

Given the physical topology of a fiber network G with n nodes m edges, and a set of wavelengths that each edge can accommodate. Also given a traffic matrix Λ and the number of transmitters and receivers at each node, our problem is for a given set P (with cardinality p) of potential lightpaths, with path p originating from node o(p) and ending at the node l(p), for each source destination nodes pair in the network, we want to determine:

(i) Which lightpaths should be chosen.

(ii) The primary path and the backup path for each such chosen lightpath, along with channel allocation.

(iii) The traffic flow pattern to meet the desired requirements based on the given traffic matrix, with the objective of minimizing the maximum congestion in the network.

We assume that only one fiber failure occurs at time and the wavelength continuous constraint (e.g., a lightpath will be assigned the same wavelength along its route) is applied.

A.2 Notations

We define the following notations employed to develop the MILP formulation.

- G [N; E] represents a physical fiber network with a set N node and a set E edge.
- N represents a set of physical nodes, with $|N| = n$.
- E represents a set of edges (bi-directional), with $|E| = m$.
- K represents a set of wavelengths that each edge $(i, j) \in E$ can accommodate.
- P represents a set of potential lightpaths.
- o(p) represents a source node of lightpath p.
- l(p) represents a destination node of lightpath p.
- $\Lambda = (\lambda^{sd})$ is a $n \times n$ static traffic matrix, where λ^{sd} represents the average traffic exchanged between each source destination pair.
- $(r^1_u, r^2_u, \ldots, r^k_u)$ is the given receiver K-vector, where r^k_u provides the number of receivers at node u tuned to channel k.
- $(t^1_u, t^2_u, \ldots, t^k_u)$ be the transmitter vector at node u, where t^k_u provides the number of transmitters tuned to channel k.
- $\varepsilon = \min\{\lambda^{sd}: \lambda^{sd} > 0, s,d \in N\}$ is the minimum value in traffic matrix.

A.3 Variables

We defined the following binary (integer) and continuous variables that used in our formulas.

Binary variables:

$$b_p = \begin{cases} 1, \text{ if lightpath p is chosen.} \\ 0, \text{ otherwise.} \end{cases} \qquad \text{for each } p \in P$$

The number of this variable is P.

$$x^p_{i,j} = \begin{cases} 1, \text{ if primary path p uses edge (i, j).} \\ 0, \text{ otherwise.} \end{cases}$$

The number of this variable is mP.

$$y^p_{i,j} = \begin{cases} 1, \text{ if backup path p uses edge (i, j).} \\ 0, \text{ otherwise.} \end{cases}$$

The number of this variable is mP.

$$w_{kp} = \begin{cases} 1, \text{ if primary path p is assigned channel k.} \\ 0, \text{ otherwise.} \end{cases}$$

The number of this variable is KP.

$$z_{kp} = \begin{cases} 1, \text{ if backup path p is assigned channel k.} \\ 0, \text{ otherwise.} \end{cases}$$

The number of this variable is KP.

Continuous variables:

δ^{kp}_{ij} = 1 iff primary path p is assigned channel k and uses edge (i, j).

The number of this variable is mKP.

α^{k}_{pq} = 1 iff both primary p and backup path q use channel k.

γ^{k}_{pq} = 1 iff backup lightpath p and backup lightpath q both use channel k.

ξ^{pq}_{ij} = 1 iff lightpath p and q share edge (i, j).

$$\theta_{pq} = \begin{cases} 1, \text{ if lightpath p and q share some edges.} \\ 0, \text{ if lightpath p and q don't share any edges.} \end{cases}$$

$$\eta^{kp}_{ij} = \begin{cases} 1, \text{ if primary path p uses edge (i, j) and corresponding backup path uses channel k.} \\ 0, \text{ otherwise.} \end{cases}$$

f^{sd}_{p} : is the amount of s-d traffic (traffic originating from s and destined for d) on path p.

F_{max} is the maximum amount of traffic flow on any lightpath.

A.4 MILP formulation and constraints

In this section we will give detailed explanation of the mathematical equations of the MILP formula and constraints. The MILP formulation consists of two parts:

-- Objective function.

-- Constraints.

Objective function:

$$\text{Minimize } F_{max}$$

As F_{max} is the maximum amount of traffic flow on any lightpath, we consider it as congestion in the network. If we can minimize the F_{max}, it is equivalent to minimize the congestion in the network.

The following section provides these constraints formulations.

A.4.1 Path creation and channel allocation constraints

$$\sum x^p_{ij} - \sum x^p_{ji} - \begin{Bmatrix} b_p, \text{ if } i = o(p) \\ -b_p, \text{ if } i = l(p) \\ 0, \text{ otherwise} \end{Bmatrix} = 0 \quad \forall\, i \in N, \forall\, p \in P \qquad (1)$$

$$\sum y^p_{ij} - \sum y^p_{ji} - \begin{Bmatrix} b_p, \text{ if } i = o(p) \\ -b_p, \text{ if } i = l(p) \\ 0, \text{ otherwise} \end{Bmatrix} = 0 \quad \forall\, i \in N, \forall\, p \in P \qquad (2)$$

Constraint (1) describes that for each node i in nodes set N, if i is the original node of a primary lightpath p, and lightpath p is chosen, then only one edge (i, j) in E is on that primary lightpath p. Constraint (2) describes the same situation but for backup lightpath.

$$\sum_{k \in K} w_{kp} - b_p = 0, \forall\, p \in P \qquad (3)$$

$$\sum_{k \in K} z_{kp} - b_p = 0, \ \forall\, p \in P \qquad (4)$$

Constraint (3) describes that if lightpath p is chosen as primary path, then it can only use one wavelength. Constraint (4) describes that if lightpath p is chosen as backup path, then it can only use one wavelength. Constraints (3) and (4) is the wavelength continuous constraint.

A.4.2 Fiber disjoint constraints

$$x^p_{ij} + y^p_{ij} - b_p \le 0, \ \ \forall (i, j) \in E, \ \ \forall\, p \in P \qquad (5)$$

This constraint describes that if edge (i, j) is used by primary lightpath p and p is chosen then it cannot be used by its corresponding backup path or visa-versa.

A.4.3 Primary paths edge-channel disjoint constraints

$$x^p_{ij} + w_{kp} - \delta^{kp}_{ij} \le 1, \ \ \forall (i, j) \in E, \ \ k \in K, \ \ p \in P \qquad (6)$$

$$\delta^{kp}_{ij} - x^p_{ij} \le 0, \ \ \forall (i, j) \in E, \ \ k \in K, \ \ p \in P \qquad (7)$$

$$\delta^{kp}_{ij} - w_{kp} \le 0, \quad \forall (i, j) \in E, \quad k \in K, \quad p \in P \tag{8}$$

$$\sum_{p=1}^{p} \delta^{kp}_{ij} \le 1, \quad \forall (i, j) \in E, \quad k \in K \tag{9}$$

Constraints (6)(7)(8) all together force the value of variable δ^{kp}_{ij} to be 1 when a primary lightpath p uses edge (i, j) and assigned wavelength k.

For example, according to definition, if primary path p uses edge (i, j), $x^{p}_{ij} = 1$ and if p uses wavelength k, $w_{kp} = 1$. From (7) and (8) we get $\delta^{kp}_{ij} \le 1$, and from (6) we get $1 \le \delta^{kp}_{ij}$, therefore $\delta^{kp}_{ij} = 1$. Constraint (9) guarantees that for all the primary lightpaths using edge (i, j), only one of them can be assigned wavelength k. Therefore provides that each primary path is edge-channel disjoint with every other primary path.

A.4.4 Primary path and other backup path edge-channel disjoint constraints

$$w_{kp} + z_{kq} - \alpha^{k}_{pq} \le 1, \quad \forall k \in K \tag{10}$$

$$- w_{kp} + \alpha^{k}_{pq} \le 0, \quad \forall k \in K \tag{11}$$

$$- z_{kq} + \alpha^{k}_{pq} \le 0, \quad \forall k \in K \tag{12}$$

$$x^{p}_{ij} + y^{q}_{ij} + \sum_{k \in K} \alpha^{k}_{pq} \le 2, \quad \forall (i, j) \in E \tag{13}$$

Constraints (10)(11)(12) all together force the value of variable α^{k}_{pq} to be 1 when a primary lightpath p uses wavelength k and a backup lightpath q uses wavelength k also.

For example, according to definition, if primary path p uses wavelength k, $w_{kp} = 1$ and if backup lightpath q uses wavelength k, $z_{kq} = 1$. From (11) and (12) we get $\alpha^k_{pq} \leq 1$, and from (10) we get $1 \leq \alpha^k_{pq}$, therefore $\alpha^k_{pq} = 1$.

Constraint (13) guarantees that if primary lightpath p and backup lightpath q both use wavelength k, then they cannot share edge (i, j). Because according to definition, if primary lightpath p and backup lightpath q both use wavelength k, then $\alpha^k_{pq} = 1$, therefore $1 \leq \Sigma\, \alpha^k_{pq}$, so x^p_{ij} and y^q_{ij} cannot be 1 at the same time, which means if primary lightpath p and backup lightpath q share edge (i, j), then they cannot be assigned the same wavelength.

A.4.5 Backup paths multiplexing constraints

$$y^p_{ij} + y^q_{ij} - \xi^{pq}_{ij} \leq 1, \quad \forall (i, j) \in E \qquad (14)$$

$$- y^p_{ij} + \xi^{pq}_{ij} \leq 0, \quad \forall (i, j) \in E \qquad (15)$$

$$- y^q_{ij} + \xi^{pq}_{ij} \leq 0, \quad \forall (i, j) \in E \qquad (16)$$

Constraints (14)(15)(16) all together force the value of variable ξ^{pq}_{ij} to be 1 when backup lightpaths p and q share edge (i, j).

For example, according to definition, if backup path p uses edge (i, j), $y^p_{ij} = 1$ and if backup lightpath q uses edge (i, j), $y^q_{ij} = 1$. From (15) and (16) we get $\xi^{pq}_{ij} \leq 1$, and from (14) we get $1 \leq \xi^{pq}_{ij}$, therefore $\xi^{pq}_{ij} = 1$. Note that here it is possible p =q.

$$\xi^{pq}_{ij} - \theta_{pq} \leq 0, \quad \forall (i, j) \in E \qquad (17)$$

$$\theta_{pq} \leq 1 \qquad (18)$$

$$\theta_{pq} - \sum_{\{(i,j)\in E\}} \xi^{pq}_{ij} \leq 0 \qquad (19)$$

Constraints (17)(18)(19) all together force the value of variable θ_{pq} to be 1 when lightpaths p and q share some edges and be 0 when they don't share any edges.

For example, according to definition, if lightpath p and q share edge (i, j), $\xi^{pq}_{ij} = 1$, from (17) we get $1 \leq \theta_{pq}$. Consider (18) we know $\theta_{pq} = 1$. If lightpaths p and q do not share any edge then $\sum\xi^{pq}_{ij}$ (i, j) will be 0, from (19) we get $\theta_{pq} \leq 0$, therefore $\theta_{pq} = 1$.

$$z_{kp} + z_{kq} - \gamma^{k}_{pq} \leq 1, \quad \forall k \in K \qquad (20)$$

$$- z_{kp} + \gamma^{k}_{pq} \leq 0, \quad \forall k \in K \qquad (21)$$

$$- z_{kq} + \gamma^{k}_{pq} \leq 0, \quad \forall k \in K \qquad (22)$$

$$x^{p}_{ij} + x^{q}_{ij} + \theta_{pq} - \sum_{\{k\in K\}} \gamma^{k}_{pq} \leq 3, \quad \forall (i,j) \in E \qquad (23)$$

Constraints (20)(21)(22) all together force the value of variable γ^{k}_{pq} to be 1 when backup lightpaths p and q both use wavelength k.

For example, according to definition, if backup lightpath p and q both use wavelength k, then $z_{kp} = 1$ and $z_{kq} = 1$. From (21) and (22) we get $\gamma^{k}_{pq} \leq 1$. From (20) we get $1 \leq \gamma^{k}_{pq}$, so $\gamma^{k}_{pq} = 1$, therefore $1 \leq \sum\gamma^{k}_{pq}$.. Consider (23), if backup lightpath p,q also share edge (i, j), according to definition, we know that $\theta_{pq} = 1$. Constraint (23) becomes:

$$x^{p}_{ij} + x^{q}_{ij} - \sum_{\{k\in K\}} \gamma^{k}_{pq} \leq 2, \quad \forall (i,j) \in E$$

as $1 \leq \Sigma\gamma^k_{pq}$, so x^p_{ij} and x^q_{ij} cannot be 1 at the same time, which means primary lightpath p and q cannot use edge (i, j) at the same time.

Constraints (14) to (23) guarantees that if two backup lightpaths share a edge and channel, then their corresponding primary lightpaths are fiber disjoint.

If lightpaths p and q do not share any edge then $1 \leq \Sigma\gamma^k_{pq}$ (i, j) will be 0, from (19) we get $\theta_{pq} \leq 0$, therefore $\theta_{pq} = 1$.

A.4.6 Constraints corresponding to transmitters and receivers

There are two cases here.

Case1: No fiber failures. In this case the following constraints must be satisfied.

$$\sum_{\{p:l(p)=u\}} w_{kp} \leq r^k_u, \quad \forall k \in K, \quad u \in N \tag{24}$$

$$\sum_{\{p:o(p)=u\}} w_{kp} \leq t^k_u, \quad \forall k \in K, \quad u \in N \tag{25}$$

Constraint (24) means the number of primary lightpaths, which destination is node u and use wavelength k, should not exceed the number of receivers at node u tuned to channel k. Constraint (25) means the number of primary lightpaths, which originate from node u and use wavelength k, should not exceed the number of transmitters at node u tuned to channel k.

Case2: If a fiber (edge) (i, j) fiber fails. In this case the following constraints must be satisfied.

$$x^{p}_{ij} + z_{kp} - \eta^{kp}_{ij} \leq 1, \quad \forall(i, j) \in E, \; k \in K, \; p \in P \tag{26}$$

$$\eta^{kp}_{ij} - x^{p}_{ij} \leq 0, \quad \forall(i, j) \in E, \; k \in K, \; p \in P \tag{27}$$

$$\eta^{kp}_{ij} - z_{kp} \leq 0, \quad \forall(i, j) \in E, \; k \in K, \; p \in P \tag{28}$$

Constraints (26)(27)(28) all together force the value of variable η^{kp}_{ij} to be 1 when primary lightpath p uses edge (i, j) and its corresponding backup lightpaths uses wavelength k.

For example, according to definition, if primary lightpath p uses edge (i, j) then $x^{p}_{ij} = 1$; if backup lightpath p uses wavelength k, then $z_{kp} = 1$, from (27) (28) we get $\eta^{kp}_{ij} \leq 1$. Consider (26), we get $1 \leq \eta^{kp}_{ij}$, therefore $\eta^{kp}_{ij} = 1$.

$$\sum_{\{p:o(p)=u\}} [w_{kp} - \delta^{kp}_{ij} + \eta^{kp}_{ij}] \leq t^{k}_{u}, \quad \forall k \in K, \quad u \in N \tag{29}$$

$$\sum_{\{p:l(p)=u\}} [w_{kp} - \delta^{kp}_{ij} + \eta^{kp}_{ij}] \leq r^{k}_{u}, \quad \forall k \in K, \quad u \in N \tag{30}$$

If edge (i, j) fails, according to definition, $\sum_{\{p:o(p)=u\}} \delta^{kp}_{ij}$ provides the number of those primary lightpaths that originated from node u, were using channel k, and are now destroyed. Similarly, $\sum_{\{p:o(p)=u\}} \eta^{kp}_{ij}$ provides the number of backup lightpaths that originated from node u, were allocated channel k, and are now replacing those destroyed primary lightpaths.

Constraint (29) means the total number of lightpaths (including the working primary paths and the backup lightpaths that replacing those destroyed primary paths) originated from node u using channel k cannot exceed the number of transmitters at node u tuned to channel k.

Constraint (30) means the total number of lightpaths (including the working primary paths and the backup lightpaths that replacing those destroyed primary paths) destined to node u using channel k cannot exceed the number of receivers at node u tuned to channel k.

A.4.7 Traffic flow constraints

$$f^{sd}_p - b_p\lambda^{sd} \le 0, \quad \forall p \in P, \quad \forall s,d \in N,\ s \mathrel{!}= d \tag{31}$$

Constraint (31) means if lightpath p is chosen, the amount of s-d traffic (traffic originating from s and destined to d) on path p cannot exceed the total amount of traffic between s and d.

$$\varepsilon b_p - \sum_{s!=d} f^{sd}_p \le 0, \quad \forall p \in P, \quad \forall s,d \in N,\ s \mathrel{!}= d \tag{32}$$

Constraint (32) means if lightpath p is chosen, then the total amount of traffic flow on lightpath p cannot be less than the smallest traffic value on the network.

$$\sum f^{sd}_p - \sum f^{sd}_p = \begin{cases} \lambda^{sd}, & \text{if } i = s \\ -\lambda^{sd}, & \text{if } i = d \\ 0, & \text{otherwise.} \end{cases} \quad \forall s,d \in N,\ s \mathrel{!}= d \tag{33}$$

Constraint (33) means for any node i, if i = s (i is the initial node), then the sum of the s-d traffic flow on all lightpaths that originated from i will be the value of λ^{sd} (the total amount of traffic between s and d). If i = d (i is the destination node), then the sum of the s-d traffic on all lightpaths that destined to i will be the value of λ^{sd}. If i is intermediate node, then the total amount of s-d traffic on all lightpaths that come into the node i will be the same as the total amount of s-d traffic on all lightpaths that go out of the node i.

$$\sum_{s!=d} f^{sd}_{p} - F_{max} \leq 0, \quad \forall p \in P \qquad (34)$$

Constraint (34) means the total amount of traffic on any lightpath cannot exceed the maxmum amount of traffic flow on any lightpath.

$$\sum_{s!=d} f^{sd}_{p} \leq 1, \quad \forall p \in P \qquad (35)$$

Constraint (35) means the total amount of traffic on any lightpath cannot be grater than 1.

Appendix B: Shortest Path Algorithm

B.1 Create three edge-disjoint shortest paths

We will illustrate our approach through the following example. Figure B.1 (a) is a six node network with 22 edges, and we want to find three shortest paths from node 0 to node 5 in this network.

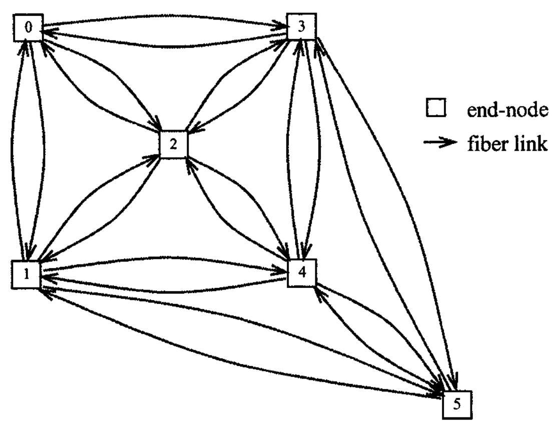


Figure B.1 (a) Six Node Network

Based on Figure B.1 (a), the first shortest path from node 0 to node 5 is $0 \rightarrow 1 \rightarrow 5$, which is found by using Dijkstra's algorithm. We modify the graph by erasing the edges used by the first shortest path, $0 \rightarrow 1$ and $1 \rightarrow 5$. The resulting graph is shown in Figure B.1 (b). Based on Figure B.1(b), we try to find the second shortest path.

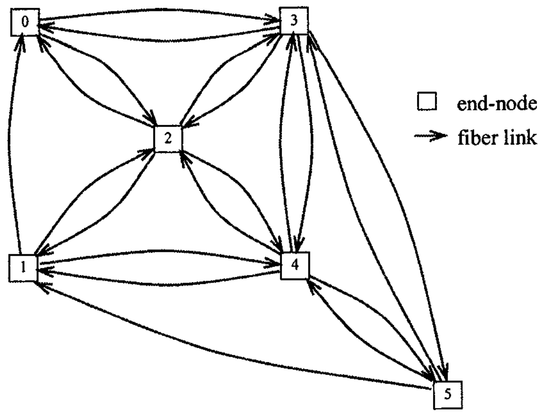


Figure B.1 (b) Six Node Network

The second shortest path from node 0 to node 5 is 0 → 3 → 5. Figure B.1 (c) shows the resulting graph after erasing the edges 0 → 3 and 3 → 5.

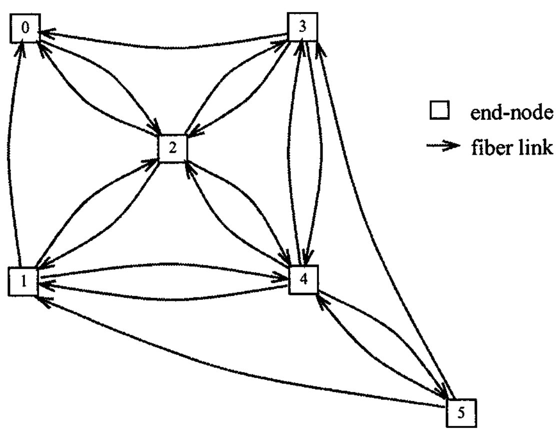


Figure B.1 (c) Six Node Network

Based on Figure B.1 (c), the third shortest path from node 0 to node 5 is 0 → 2 → 4 → 5.

B.2 Dijkstra's algorithm

Dijkstra's algorithm finds shortest paths from *source* to all the other nodes in the network. Dijkstra's algorithm maintains a distance label *label*[*i*] with each node *i*, which is an upper bound on the shortest path length to node *i*. At any intermediate step, the algorithm divides the nodes into two groups: those it designates as permanently labeled (or permanent) and those it designates as temporarily labeled (or temporary). The distance label to any permanent node represents the shortest distance from the source to that node. For any temporary node, the distance label is an upper bound on the shortest path distance to that node.

The basic idea of the algorithm is to change temporary nodes to permanent nodes by updating distance labels. Initially, we give node *source* a permanent label of 0, and each other node a temporary label equal to ∞. At each iteration, the label of a node is its shortest distance from the source node along a path whose internal nodes are all permanently labeled. The algorithm selects a node with the minimum temporary label, makes it permanent, and reaches out from that node, to update the distance labels of adjacent nodes. The algorithm terminates when it has designated all nodes as permanent. The pseudocode of Dijkstra's algorithm is shown in Figure B.2.

Step1: create *permanent* list containing only *source*

Step2: create *temporary* list containing all nodes except *source*

Step3: for each node *i* in the network, do Step 4 – Step 5

Step4: the distance from *source* to node *i* is 99999

Step5: if there exists an edge from *source* to node *i*, then

  the distance from *source* to node *i* is 1, and

  the preceding node of node *i* is *source*

Step6: the distance from *source* to *source* is 0, and the preceding node of *source* is *source*

Step7: while *temporary* list isn't empty, do Step 8 – Step 10

Step8: if node *j* is the node with the smallest distance from source in *temporary* list, then

  add node *j* to *permanent* list, and

  delete node *j* from *temporary* list

Step9: for each node *k* in the network, do Step 10

Step10: if there exists an edge from node *j* to node *k* and the distance from *source* to node *k* is greater than (1 + the distance from *source* to node *j*), then

  the distance from *source* to node *k* =

    1 + the distance from *source* to node *j*

  the preceding node of node *k* is node *j*

Step11: if *destination* has a preceding node, then a shortest path is found

Figure B.2 Dijkstra's Algorithm

Appendix C: Traffic Matrices

The traffic matrices for the experiments with a six node network (section 5.1) are shown below:

$$T_1 = \begin{bmatrix} 0 & 10 & 20 & 30 & 60 & 10 \\ 10 & 0 & 20 & 10 & 10 & 70 \\ 90 & 10 & 0 & 30 & 10 & 10 \\ 10 & 20 & 80 & 0 & 30 & 20 \\ 10 & 30 & 10 & 60 & 0 & 10 \\ 20 & 10 & 20 & 10 & 80 & 0 \end{bmatrix} \quad T_2 = \begin{bmatrix} 0 & 90 & 20 & 10 & 20 & 40 \\ 30 & 0 & 10 & 20 & 30 & 70 \\ 80 & 20 & 0 & 10 & 10 & 30 \\ 10 & 20 & 60 & 0 & 30 & 90 \\ 10 & 40 & 10 & 95 & 0 & 10 \\ 20 & 40 & 20 & 10 & 60 & 0 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} 0 & 60 & 20 & 30 & 70 & 10 \\ 50 & 0 & 10 & 60 & 50 & 70 \\ 40 & 70 & 0 & 30 & 10 & 50 \\ 70 & 20 & 60 & 0 & 30 & 20 \\ 10 & 40 & 10 & 70 & 0 & 50 \\ 20 & 50 & 40 & 10 & 90 & 0 \end{bmatrix} \quad T_4 = \begin{bmatrix} 0 & 90 & 20 & 50 & 70 & 40 \\ 10 & 0 & 40 & 80 & 50 & 70 \\ 50 & 40 & 0 & 30 & 40 & 90 \\ 70 & 40 & 60 & 0 & 50 & 20 \\ 80 & 40 & 50 & 20 & 0 & 60 \\ 20 & 50 & 40 & 60 & 90 & 0 \end{bmatrix}$$

$$T_5 = \begin{bmatrix} 0 & 90 & 60 & 20 & 70 & 80 \\ 60 & 0 & 10 & 80 & 50 & 70 \\ 80 & 60 & 0 & 70 & 10 & 90 \\ 70 & 20 & 90 & 0 & 80 & 60 \\ 80 & 80 & 60 & 20 & 0 & 90 \\ 20 & 50 & 70 & 80 & 90 & 0 \end{bmatrix} \quad T_6 = \begin{bmatrix} 0 & 80 & 60 & 90 & 70 & 85 \\ 90 & 0 & 75 & 80 & 65 & 70 \\ 80 & 70 & 0 & 60 & 90 & 60 \\ 70 & 95 & 90 & 0 & 80 & 55 \\ 80 & 60 & 85 & 70 & 0 & 90 \\ 95 & 85 & 80 & 85 & 90 & 0 \end{bmatrix}$$

$$T_7 = \begin{bmatrix} 0 & 100 & 100 & 100 & 100 & 100 \\ 100 & 0 & 100 & 100 & 100 & 100 \\ 100 & 100 & 0 & 100 & 100 & 100 \\ 100 & 100 & 100 & 0 & 100 & 100 \\ 100 & 100 & 100 & 100 & 0 & 100 \\ 100 & 100 & 100 & 100 & 100 & 0 \end{bmatrix}$$

The traffic matrices for the experiments with a fourteen node network (section 5.2) are shown below:

$$T_1 = \begin{bmatrix} 0 & 30 & 20 & 30 & 10 & 10 & 80 & 30 & 40 & 10 & 10 & 10 & 10 & 10 \\ 10 & 0 & 20 & 10 & 30 & 10 & 10 & 60 & 10 & 10 & 50 & 10 & 30 & 10 \\ 40 & 10 & 0 & 50 & 10 & 20 & 30 & 10 & 30 & 10 & 10 & 20 & 10 & 80 \\ 10 & 40 & 10 & 0 & 60 & 10 & 20 & 30 & 10 & 30 & 10 & 50 & 20 & 10 \\ 20 & 90 & 10 & 50 & 0 & 60 & 40 & 10 & 10 & 10 & 20 & 30 & 20 & 10 \\ 40 & 10 & 30 & 10 & 10 & 0 & 50 & 60 & 20 & 35 & 10 & 15 & 90 & 10 \\ 10 & 20 & 20 & 30 & 10 & 40 & 0 & 10 & 70 & 20 & 40 & 10 & 20 & 10 \\ 90 & 10 & 50 & 15 & 30 & 10 & 20 & 0 & 20 & 10 & 45 & 10 & 10 & 30 \\ 10 & 10 & 40 & 10 & 20 & 50 & 20 & 10 & 0 & 80 & 20 & 30 & 10 & 10 \\ 50 & 30 & 10 & 40 & 80 & 10 & 25 & 10 & 20 & 0 & 30 & 10 & 20 & 10 \\ 15 & 10 & 60 & 20 & 10 & 20 & 20 & 10 & 80 & 10 & 0 & 45 & 10 & 30 \\ 10 & 70 & 10 & 20 & 30 & 20 & 10 & 85 & 20 & 25 & 40 & 0 & 10 & 10 \\ 20 & 30 & 40 & 10 & 80 & 30 & 10 & 10 & 10 & 25 & 10 & 90 & 0 & 15 \\ 30 & 10 & 20 & 10 & 40 & 20 & 10 & 10 & 50 & 10 & 30 & 10 & 70 & 0 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 0 & 30 & 20 & 30 & 10 & 10 & 80 & 30 & 40 & 10 & 10 & 10 & 10 & 10 \\ 10 & 0 & 20 & 10 & 30 & 10 & 10 & 60 & 10 & 10 & 50 & 10 & 30 & 10 \\ 40 & 10 & 0 & 50 & 10 & 20 & 30 & 10 & 30 & 10 & 10 & 20 & 10 & 80 \\ 10 & 40 & 10 & 0 & 60 & 10 & 20 & 30 & 10 & 30 & 10 & 50 & 20 & 10 \\ 20 & 90 & 10 & 50 & 0 & 60 & 40 & 10 & 10 & 10 & 20 & 30 & 20 & 10 \\ 40 & 10 & 30 & 10 & 10 & 0 & 50 & 60 & 20 & 35 & 10 & 15 & 90 & 10 \\ 10 & 20 & 20 & 30 & 10 & 40 & 0 & 10 & 70 & 20 & 40 & 10 & 20 & 10 \\ 90 & 10 & 50 & 15 & 30 & 10 & 20 & 0 & 20 & 10 & 45 & 10 & 10 & 30 \\ 10 & 10 & 40 & 10 & 20 & 50 & 20 & 10 & 0 & 80 & 20 & 30 & 10 & 10 \\ 50 & 30 & 10 & 40 & 80 & 10 & 25 & 10 & 20 & 0 & 30 & 10 & 20 & 10 \\ 15 & 10 & 60 & 20 & 10 & 20 & 20 & 10 & 80 & 10 & 0 & 45 & 10 & 30 \\ 10 & 70 & 10 & 20 & 30 & 20 & 10 & 85 & 20 & 25 & 40 & 0 & 10 & 10 \\ 20 & 30 & 40 & 10 & 80 & 30 & 10 & 10 & 10 & 25 & 10 & 90 & 0 & 15 \\ 30 & 10 & 20 & 10 & 40 & 20 & 10 & 10 & 50 & 10 & 30 & 10 & 70 & 0 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} 0 & 30 & 80 & 30 & 40 & 10 & 50 & 30 & 40 & 10 & 50 & 50 & 10 & 60 \\ 50 & 0 & 10 & 70 & 10 & 60 & 10 & 50 & 30 & 40 & 20 & 50 & 40 & 10 \\ 40 & 80 & 0 & 50 & 40 & 50 & 30 & 50 & 20 & 10 & 60 & 20 & 40 & 30 \\ 50 & 10 & 60 & 0 & 30 & 40 & 10 & 40 & 50 & 30 & 50 & 70 & 10 & 40 \\ 10 & 40 & 30 & 50 & 0 & 80 & 20 & 10 & 50 & 30 & 70 & 60 & 40 & 50 \\ 80 & 20 & 50 & 60 & 30 & 0 & 70 & 40 & 35 & 50 & 20 & 40 & 50 & 10 \\ 30 & 50 & 40 & 10 & 50 & 40 & 0 & 20 & 50 & 60 & 10 & 50 & 70 & 30 \\ 50 & 10 & 45 & 95 & 40 & 30 & 60 & 0 & 40 & 10 & 70 & 10 & 40 & 50 \\ 10 & 40 & 50 & 20 & 60 & 50 & 20 & 80 & 0 & 40 & 50 & 60 & 10 & 40 \\ 60 & 50 & 10 & 80 & 10 & 60 & 40 & 30 & 50 & 0 & 40 & 50 & 30 & 40 \\ 45 & 30 & 50 & 10 & 60 & 10 & 40 & 60 & 80 & 50 & 0 & 10 & 40 & 50 \\ 10 & 70 & 30 & 50 & 40 & 40 & 10 & 30 & 40 & 95 & 40 & 0 & 60 & 10 \\ 50 & 10 & 40 & 10 & 80 & 10 & 30 & 45 & 10 & 60 & 50 & 50 & 0 & 90 \\ 30 & 80 & 10 & 50 & 40 & 50 & 40 & 20 & 70 & 40 & 40 & 50 & 30 & 0 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} 0 & 30 & 60 & 80 & 50 & 10 & 70 & 60 & 40 & 10 & 60 & 70 & 10 & 90 \\ 75 & 0 & 50 & 70 & 40 & 60 & 10 & 90 & 50 & 50 & 40 & 50 & 60 & 50 \\ 60 & 70 & 0 & 50 & 60 & 75 & 40 & 80 & 40 & 60 & 60 & 40 & 40 & 70 \\ 40 & 10 & 60 & 0 & 80 & 50 & 10 & 60 & 70 & 90 & 50 & 70 & 60 & 40 \\ 30 & 70 & 55 & 90 & 0 & 60 & 50 & 10 & 50 & 60 & 70 & 60 & 50 & 55 \\ 80 & 60 & 50 & 60 & 80 & 0 & 70 & 40 & 35 & 50 & 40 & 40 & 70 & 40 \\ 50 & 50 & 60 & 60 & 70 & 40 & 0 & 80 & 70 & 50 & 60 & 50 & 90 & 65 \\ 60 & 60 & 45 & 55 & 80 & 30 & 90 & 0 & 40 & 60 & 60 & 40 & 70 & 50 \\ 10 & 40 & 70 & 40 & 60 & 50 & 40 & 80 & 0 & 50 & 50 & 60 & 50 & 40 \\ 60 & 60 & 50 & 90 & 50 & 60 & 50 & 60 & 80 & 0 & 60 & 10 & 60 & 70 \\ 45 & 50 & 50 & 30 & 60 & 50 & 80 & 60 & 60 & 90 & 0 & 70 & 30 & 40 \\ 40 & 70 & 60 & 80 & 50 & 60 & 40 & 70 & 40 & 95 & 40 & 0 & 50 & 60 \\ 60 & 50 & 40 & 50 & 90 & 40 & 75 & 10 & 90 & 70 & 60 & 50 & 0 & 50 \\ 40 & 60 & 75 & 60 & 40 & 50 & 60 & 80 & 50 & 60 & 50 & 60 & 40 & 0 \end{bmatrix}$$

$$T_5 = \begin{bmatrix} 0 & 50 & 70 & 80 & 60 & 40 & 70 & 60 & 85 & 70 & 60 & 70 & 30 & 90 \\ 75 & 0 & 65 & 70 & 40 & 60 & 80 & 90 & 60 & 40 & 70 & 50 & 60 & 30 \\ 60 & 70 & 0 & 90 & 70 & 85 & 60 & 20 & 75 & 60 & 60 & 80 & 40 & 60 \\ 90 & 75 & 50 & 0 & 80 & 55 & 80 & 60 & 75 & 90 & 50 & 70 & 60 & 80 \\ 50 & 70 & 80 & 90 & 0 & 60 & 50 & 80 & 70 & 65 & 70 & 60 & 75 & 55 \\ 80 & 70 & 50 & 60 & 90 & 0 & 70 & 50 & 65 & 25 & 80 & 90 & 70 & 60 \\ 65 & 70 & 50 & 60 & 75 & 85 & 0 & 80 & 90 & 60 & 70 & 60 & 90 & 50 \\ 70 & 90 & 75 & 85 & 50 & 70 & 90 & 0 & 40 & 75 & 60 & 80 & 70 & 70 \\ 80 & 40 & 70 & 60 & 65 & 95 & 50 & 70 & 0 & 80 & 70 & 60 & 90 & 20 \\ 60 & 80 & 50 & 90 & 70 & 60 & 90 & 55 & 80 & 0 & 70 & 60 & 80 & 70 \\ 55 & 70 & 90 & 60 & 70 & 50 & 80 & 65 & 60 & 90 & 0 & 70 & 80 & 40 \\ 75 & 70 & 60 & 80 & 90 & 20 & 40 & 70 & 80 & 95 & 60 & 0 & 90 & 65 \\ 85 & 60 & 90 & 75 & 80 & 60 & 75 & 40 & 90 & 50 & 60 & 80 & 0 & 70 \\ 50 & 80 & 75 & 60 & 95 & 70 & 60 & 80 & 50 & 70 & 80 & 60 & 90 & 0 \end{bmatrix}$$

$$T_6 = \begin{bmatrix} 0 & 70 & 90 & 80 & 50 & 95 & 90 & 60 & 70 & 90 & 85 & 70 & 80 & 90 \\ 95 & 0 & 85 & 95 & 85 & 90 & 80 & 90 & 80 & 85 & 90 & 95 & 70 & 60 \\ 80 & 90 & 0 & 90 & 90 & 85 & 95 & 80 & 90 & 85 & 90 & 80 & 70 & 90 \\ 90 & 80 & 70 & 0 & 80 & 90 & 80 & 90 & 75 & 90 & 80 & 70 & 90 & 80 \\ 85 & 70 & 90 & 90 & 0 & 80 & 70 & 80 & 95 & 90 & 70 & 80 & 95 & 95 \\ 80 & 90 & 90 & 80 & 90 & 0 & 90 & 70 & 95 & 85 & 80 & 90 & 95 & 80 \\ 90 & 70 & 80 & 90 & 70 & 85 & 0 & 90 & 90 & 70 & 00 & 80 & 90 & 90 \\ 60 & 90 & 95 & 85 & 90 & 70 & 90 & 0 & 80 & 85 & 90 & 80 & 70 & 75 \\ 80 & 85 & 90 & 90 & 85 & 95 & 70 & 80 & 0 & 90 & 85 & 95 & 80 & 90 \\ 60 & 80 & 80 & 90 & 95 & 90 & 80 & 90 & 90 & 0 & 90 & 80 & 90 & 90 \\ 95 & 80 & 90 & 80 & 90 & 85 & 80 & 85 & 60 & 90 & 0 & 70 & 80 & 95 \\ 75 & 90 & 95 & 80 & 90 & 90 & 90 & 70 & 80 & 95 & 80 & 0 & 90 & 80 \\ 80 & 95 & 90 & 75 & 80 & 90 & 85 & 90 & 90 & 30 & 70 & 80 & 0 & 90 \\ 95 & 80 & 95 & 90 & 90 & 70 & 90 & 80 & 80 & 90 & 85 & 95 & 90 & 0 \end{bmatrix}$$

$$T_7 = \begin{bmatrix} 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 \\ 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 \end{bmatrix}$$

The traffic matrices for the experiments with a twenty node network (section 5.3) are shown below:

$$T_1 = \begin{bmatrix}
0 & 30 & 30 & 30 & 10 & 10 & 80 & 30 & 40 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 20 & 10 & 10 \\
10 & 0 & 10 & 10 & 10 & 10 & 10 & 60 & 10 & 10 & 10 & 10 & 10 & 10 & 40 & 10 & 10 & 10 & 10 & 20 \\
40 & 10 & 0 & 10 & 10 & 10 & 30 & 10 & 10 & 10 & 60 & 20 & 10 & 10 & 10 & 70 & 10 & 10 & 30 & 10 \\
10 & 10 & 10 & 0 & 10 & 10 & 10 & 80 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 40 & 10 & 10 & 10 \\
10 & 90 & 10 & 50 & 0 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 20 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 0 & 10 & 10 & 10 & 10 & 10 & 10 & 90 & 10 & 10 & 10 & 10 & 50 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 0 & 10 & 70 & 60 & 10 & 10 & 70 & 10 & 10 & 10 & 10 & 10 & 20 & 10 \\
10 & 10 & 10 & 95 & 10 & 10 & 10 & 0 & 10 & 10 & 10 & 10 & 10 & 80 & 10 & 10 & 10 & 10 & 10 & 40 \\
10 & 10 & 10 & 10 & 10 & 50 & 20 & 10 & 0 & 10 & 50 & 60 & 10 & 10 & 10 & 30 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 0 & 30 & 10 & 10 & 70 & 10 & 10 & 90 & 10 & 10 & 10 \\
15 & 10 & 10 & 10 & 10 & 10 & 20 & 10 & 10 & 10 & 0 & 10 & 10 & 10 & 80 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 95 & 40 & 0 & 10 & 10 & 10 & 10 & 10 & 30 & 10 & 10 \\
20 & 10 & 10 & 10 & 80 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 0 & 10 & 10 & 50 & 10 & 10 & 30 & 10 \\
30 & 10 & 10 & 10 & 40 & 50 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 0 & 10 & 10 & 20 & 10 & 10 & 70 \\
10 & 40 & 10 & 10 & 10 & 10 & 10 & 20 & 10 & 10 & 10 & 10 & 10 & 50 & 0 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 95 & 10 & 10 & 10 & 10 & 10 & 30 & 10 & 10 & 10 & 10 & 10 & 10 & 0 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 50 & 10 & 10 & 10 & 10 & 0 & 70 & 10 & 10 \\
10 & 10 & 10 & 70 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 20 & 10 & 10 & 10 & 0 & 10 & 20 \\
10 & 10 & 10 & 10 & 10 & 40 & 20 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 0 & 10 \\
70 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 0 & 1 & 10 & 10 & 10 & 10 & 20 & 10 & 10 & 10 & 10 & 0
\end{bmatrix}
\qquad
T_2 = \begin{bmatrix}
0 & 30 & 20 & 30 & 10 & 10 & 80 & 30 & 40 & 10 & 10 & 10 & 10 & 10 & 20 & 30 & 10 & 10 & 20 & 30 \\
10 & 0 & 20 & 10 & 30 & 10 & 10 & 60 & 10 & 10 & 50 & 10 & 30 & 10 & 10 & 20 & 10 & 30 & 10 & 10 \\
40 & 10 & 0 & 50 & 10 & 20 & 30 & 10 & 30 & 10 & 10 & 20 & 10 & 80 & 20 & 40 & 10 & 10 & 50 & 20 \\
10 & 40 & 10 & 0 & 60 & 10 & 20 & 30 & 10 & 30 & 10 & 50 & 20 & 10 & 30 & 10 & 50 & 30 & 10 & 10 \\
20 & 90 & 10 & 50 & 0 & 60 & 40 & 10 & 10 & 10 & 20 & 30 & 20 & 10 & 10 & 30 & 20 & 10 & 40 & 10 \\
40 & 10 & 30 & 10 & 10 & 0 & 50 & 60 & 20 & 35 & 10 & 15 & 90 & 10 & 20 & 40 & 10 & 10 & 50 & 20 \\
10 & 20 & 20 & 30 & 10 & 40 & 0 & 10 & 70 & 20 & 40 & 10 & 20 & 10 & 10 & 30 & 50 & 20 & 10 & 30 \\
90 & 10 & 50 & 15 & 30 & 10 & 20 & 0 & 20 & 10 & 45 & 10 & 10 & 30 & 20 & 10 & 10 & 30 & 40 & 10 \\
10 & 10 & 40 & 10 & 20 & 50 & 20 & 10 & 0 & 80 & 20 & 30 & 10 & 10 & 30 & 50 & 10 & 40 & 10 & 10 \\
50 & 30 & 10 & 40 & 80 & 10 & 25 & 10 & 20 & 0 & 30 & 10 & 20 & 10 & 50 & 10 & 30 & 10 & 10 & 40 \\
15 & 10 & 60 & 20 & 10 & 20 & 20 & 10 & 80 & 10 & 0 & 45 & 10 & 30 & 10 & 20 & 40 & 10 & 50 & 10 \\
10 & 70 & 10 & 20 & 30 & 20 & 10 & 85 & 20 & 25 & 40 & 0 & 10 & 10 & 60 & 30 & 10 & 10 & 20 & 10 \\
20 & 30 & 40 & 10 & 80 & 30 & 10 & 10 & 10 & 25 & 10 & 90 & 0 & 15 & 10 & 40 & 10 & 50 & 10 & 10 \\
30 & 10 & 20 & 10 & 40 & 20 & 10 & 10 & 50 & 10 & 30 & 10 & 70 & 0 & 10 & 10 & 20 & 10 & 10 & 70 \\
10 & 40 & 10 & 50 & 10 & 10 & 10 & 20 & 10 & 10 & 10 & 10 & 10 & 50 & 0 & 10 & 10 & 30 & 40 & 10 \\
10 & 10 & 95 & 10 & 10 & 10 & 10 & 10 & 30 & 10 & 60 & 10 & 30 & 10 & 10 & 0 & 10 & 10 & 10 & 30 \\
10 & 10 & 10 & 20 & 40 & 10 & 30 & 10 & 10 & 10 & 10 & 50 & 10 & 10 & 30 & 10 & 0 & 70 & 30 & 10 \\
10 & 40 & 10 & 70 & 10 & 10 & 10 & 70 & 10 & 40 & 10 & 10 & 10 & 20 & 10 & 10 & 10 & 0 & 10 & 20 \\
20 & 90 & 10 & 50 & 0 & 60 & 40 & 10 & 10 & 10 & 20 & 30 & 20 & 10 & 10 & 30 & 20 & 10 & 0 & 10 \\
70 & 10 & 10 & 20 & 10 & 10 & 50 & 10 & 10 & 30 & 10 & 10 & 40 & 10 & 20 & 10 & 30 & 10 & 10 & 0
\end{bmatrix}$$

$$T_3 = \begin{bmatrix}
0\ 30\ 80\ 30\ 40\ 10\ 50\ 30\ 40\ 10\ 50\ 50\ 10\ 60\ 50\ 70\ 40\ 75\ 60\ 80 \\
50\ 0\ 10\ 70\ 10\ 60\ 10\ 50\ 30\ 40\ 20\ 10\ 40\ 10\ 40\ 60\ 50\ 80\ 30\ 40 \\
40\ 80\ 0\ 50\ 40\ 50\ 30\ 50\ 10\ 10\ 10\ 20\ 40\ 30\ 10\ 10\ 60\ 70\ 10\ 10 \\
50\ 10\ 60\ 0\ 30\ 40\ 10\ 40\ 10\ 30\ 50\ 10\ 10\ 40\ 60\ 30\ 10\ 40\ 60\ 50 \\
10\ 10\ 30\ 50\ 0\ 80\ 20\ 10\ 50\ 30\ 10\ 60\ 40\ 50\ 70\ 10\ 10\ 60\ 10\ 30 \\
80\ 20\ 50\ 60\ 30\ 0\ 70\ 40\ 15\ 50\ 20\ 40\ 50\ 10\ 30\ 40\ 60\ 40\ 30\ 50 \\
30\ 50\ 40\ 10\ 50\ 40\ 0\ 20\ 50\ 60\ 10\ 50\ 70\ 30\ 10\ 50\ 30\ 10\ 60\ 10 \\
50\ 10\ 15\ 95\ 40\ 30\ 60\ 0\ 40\ 10\ 70\ 10\ 40\ 50\ 40\ 70\ 50\ 40\ 10\ 50 \\
10\ 40\ 50\ 20\ 10\ 50\ 20\ 80\ 0\ 40\ 50\ 60\ 10\ 10\ 50\ 30\ 10\ 10\ 40\ 30 \\
60\ 50\ 10\ 80\ 10\ 60\ 40\ 30\ 50\ 0\ 10\ 50\ 10\ 30\ 10\ 40\ 50\ 30\ 20\ 70 \\
45\ 30\ 50\ 10\ 60\ 10\ 40\ 10\ 20\ 50\ 0\ 10\ 40\ 50\ 60\ 40\ 10\ 30\ 10\ 50 \\
10\ 70\ 30\ 50\ 40\ 40\ 10\ 30\ 40\ 95\ 40\ 0\ 60\ 10\ 40\ 50\ 30\ 40\ 50\ 60 \\
50\ 10\ 40\ 10\ 80\ 10\ 30\ 45\ 10\ 60\ 50\ 50\ 0\ 90\ 30\ 60\ 40\ 50\ 30\ 20 \\
30\ 80\ 10\ 50\ 40\ 50\ 40\ 20\ 30\ 40\ 40\ 50\ 30\ 0\ 10\ 10\ 20\ 10\ 10\ 70 \\
10\ 40\ 10\ 30\ 10\ 20\ 10\ 50\ 10\ 20\ 10\ 30\ 10\ 50\ 0\ 10\ 40\ 10\ 20\ 10 \\
10\ 10\ 95\ 10\ 30\ 50\ 10\ 10\ 30\ 10\ 40\ 10\ 30\ 40\ 10\ 0\ 10\ 50\ 10\ 40 \\
30\ 10\ 10\ 20\ 10\ 10\ 30\ 40\ 10\ 30\ 10\ 50\ 10\ 10\ 30\ 10\ 0\ 70\ 30\ 10 \\
10\ 40\ 30\ 70\ 40\ 30\ 10\ 40\ 20\ 10\ 40\ 10\ 20\ 20\ 10\ 30\ 10\ 0\ 10\ 20 \\
10\ 10\ 50\ 10\ 10\ 40\ 20\ 10\ 10\ 30\ 10\ 50\ 10\ 10\ 50\ 10\ 10\ 30\ 0\ 10 \\
70\ 10\ 40\ 10\ 20\ 10\ 10\ 30\ 10\ 01\ 10\ 10\ 30\ 10\ 20\ 10\ 30\ 10\ 50\ 0
\end{bmatrix}
\quad
T_4 = \begin{bmatrix}
0\ 30\ 80\ 30\ 40\ 10\ 50\ 30\ 40\ 10\ 50\ 50\ 10\ 60\ 50\ 70\ 40\ 75\ 60\ 80 \\
50\ 0\ 10\ 70\ 10\ 60\ 10\ 50\ 30\ 40\ 20\ 50\ 40\ 10\ 40\ 60\ 50\ 80\ 30\ 40 \\
40\ 80\ 0\ 50\ 40\ 50\ 30\ 50\ 20\ 10\ 60\ 20\ 40\ 30\ 50\ 30\ 60\ 70\ 10\ 40 \\
50\ 10\ 60\ 0\ 30\ 40\ 10\ 40\ 50\ 30\ 50\ 70\ 10\ 40\ 60\ 30\ 50\ 40\ 60\ 50 \\
10\ 40\ 30\ 50\ 0\ 80\ 20\ 10\ 50\ 30\ 70\ 60\ 40\ 50\ 70\ 30\ 40\ 60\ 50\ 30 \\
80\ 20\ 50\ 60\ 30\ 0\ 70\ 40\ 35\ 50\ 20\ 40\ 50\ 10\ 30\ 40\ 60\ 40\ 30\ 50 \\
30\ 50\ 40\ 10\ 50\ 40\ 0\ 20\ 50\ 60\ 10\ 50\ 70\ 30\ 60\ 50\ 30\ 50\ 60\ 20 \\
50\ 10\ 45\ 95\ 40\ 30\ 60\ 0\ 40\ 10\ 70\ 10\ 40\ 50\ 40\ 70\ 50\ 40\ 30\ 50 \\
10\ 40\ 50\ 20\ 60\ 50\ 20\ 80\ 0\ 40\ 50\ 60\ 10\ 40\ 50\ 30\ 60\ 50\ 40\ 30 \\
60\ 50\ 10\ 80\ 10\ 60\ 40\ 30\ 50\ 0\ 40\ 50\ 30\ 40\ 60\ 40\ 50\ 30\ 20\ 70 \\
45\ 30\ 50\ 10\ 60\ 10\ 40\ 60\ 80\ 50\ 0\ 10\ 40\ 50\ 60\ 40\ 30\ 60\ 30\ 50 \\
10\ 70\ 30\ 50\ 40\ 40\ 10\ 30\ 40\ 95\ 40\ 0\ 60\ 10\ 40\ 50\ 30\ 40\ 50\ 60 \\
50\ 10\ 40\ 10\ 80\ 10\ 30\ 45\ 10\ 60\ 50\ 50\ 0\ 90\ 30\ 60\ 40\ 50\ 30\ 20 \\
30\ 80\ 10\ 50\ 40\ 50\ 40\ 20\ 70\ 40\ 40\ 50\ 30\ 0\ 10\ 10\ 20\ 10\ 10\ 70 \\
10\ 40\ 10\ 20\ 10\ 30\ 10\ 20\ 10\ 30\ 10\ 20\ 10\ 50\ 0\ 30\ 10\ 40\ 10\ 10 \\
10\ 10\ 95\ 10\ 30\ 10\ 20\ 10\ 30\ 10\ 50\ 10\ 30\ 10\ 10\ 0\ 10\ 10\ 50\ 10 \\
30\ 50\ 40\ 10\ 50\ 40\ 20\ 20\ 50\ 60\ 10\ 50\ 70\ 30\ 60\ 50\ 0\ 50\ 60\ 20 \\
50\ 10\ 45\ 95\ 40\ 30\ 60\ 50\ 40\ 10\ 70\ 10\ 40\ 50\ 40\ 70\ 50\ 0\ 30\ 50 \\
10\ 40\ 50\ 20\ 60\ 50\ 20\ 80\ 50\ 40\ 50\ 60\ 10\ 40\ 50\ 30\ 60\ 50\ 0\ 30 \\
70\ 10\ 10\ 30\ 10\ 40\ 30\ 60\ 10\ 30\ 10\ 80\ 10\ 30\ 20\ 10\ 30\ 40\ 10\ 0
\end{bmatrix}$$

$$
T_5 = \begin{bmatrix}
0 & 30 & 60 & 80 & 50 & 10 & 70 & 60 & 40 & 10 & 60 & 70 & 10 & 90 & 30 & 60 & 80 & 50 & 10 & 70 \\
75 & 0 & 50 & 70 & 40 & 60 & 10 & 90 & 50 & 50 & 40 & 50 & 60 & 50 & 50 & 70 & 40 & 60 & 10 & 90 \\
60 & 70 & 0 & 50 & 60 & 75 & 40 & 80 & 40 & 60 & 60 & 40 & 40 & 70 & 50 & 60 & 75 & 40 & 80 & 40 \\
40 & 10 & 60 & 0 & 80 & 50 & 10 & 60 & 70 & 90 & 50 & 70 & 60 & 40 & 80 & 50 & 10 & 60 & 70 & 90 \\
30 & 70 & 55 & 90 & 0 & 60 & 50 & 10 & 50 & 60 & 70 & 60 & 50 & 55 & 60 & 50 & 10 & 50 & 60 & 70 \\
80 & 60 & 50 & 60 & 80 & 0 & 70 & 40 & 35 & 50 & 40 & 40 & 70 & 40 & 70 & 40 & 35 & 50 & 40 & 40 \\
50 & 50 & 60 & 60 & 70 & 40 & 0 & 80 & 70 & 50 & 60 & 50 & 90 & 65 & 80 & 70 & 50 & 60 & 50 & 90 \\
60 & 60 & 45 & 55 & 80 & 30 & 90 & 0 & 40 & 60 & 60 & 40 & 70 & 50 & 60 & 60 & 45 & 55 & 80 & 30 \\
10 & 40 & 70 & 40 & 60 & 50 & 40 & 80 & 0 & 50 & 50 & 60 & 50 & 40 & 50 & 50 & 30 & 60 & 50 & 80 \\
60 & 60 & 50 & 90 & 50 & 60 & 50 & 60 & 80 & 0 & 60 & 10 & 60 & 70 & 40 & 50 & 90 & 40 & 75 & 10 \\
45 & 50 & 50 & 30 & 60 & 50 & 80 & 60 & 60 & 90 & 0 & 70 & 30 & 40 & 50 & 60 & 40 & 70 & 40 & 95 \\
40 & 70 & 60 & 80 & 50 & 60 & 40 & 70 & 40 & 95 & 40 & 0 & 50 & 60 & 60 & 40 & 50 & 60 & 80 & 50 \\
60 & 50 & 40 & 50 & 90 & 40 & 75 & 10 & 90 & 70 & 60 & 50 & 0 & 50 & 60 & 45 & 55 & 80 & 30 & 90 \\
40 & 60 & 75 & 60 & 40 & 50 & 60 & 80 & 50 & 60 & 50 & 60 & 40 & 0 & 40 & 75 & 40 & 90 & 70 & 60 \\
10 & 40 & 10 & 20 & 60 & 30 & 70 & 20 & 10 & 30 & 10 & 20 & 60 & 50 & 0 & 30 & 10 & 40 & 10 & 40 \\
10 & 10 & 95 & 50 & 30 & 50 & 20 & 60 & 30 & 10 & 50 & 10 & 30 & 10 & 50 & 0 & 40 & 10 & 50 & 10 \\
30 & 50 & 40 & 10 & 50 & 40 & 20 & 20 & 50 & 60 & 10 & 50 & 70 & 30 & 60 & 50 & 0 & 50 & 60 & 20 \\
50 & 10 & 45 & 95 & 40 & 30 & 60 & 50 & 40 & 10 & 70 & 10 & 40 & 50 & 40 & 70 & 50 & 0 & 30 & 50 \\
10 & 40 & 50 & 20 & 60 & 50 & 20 & 80 & 50 & 40 & 50 & 60 & 10 & 40 & 50 & 30 & 60 & 50 & 0 & 30 \\
70 & 50 & 10 & 30 & 10 & 40 & 30 & 60 & 10 & 30 & 10 & 80 & 70 & 30 & 20 & 10 & 30 & 40 & 50 & 0
\end{bmatrix}
$$

$$
T_6 = \begin{bmatrix}
0 & 30 & 60 & 80 & 50 & 10 & 70 & 60 & 40 & 10 & 60 & 70 & 10 & 90 & 30 & 60 & 80 & 50 & 10 & 70 \\
75 & 0 & 50 & 70 & 40 & 60 & 10 & 90 & 50 & 50 & 40 & 50 & 60 & 50 & 50 & 70 & 40 & 60 & 10 & 90 \\
60 & 70 & 0 & 50 & 60 & 75 & 40 & 80 & 40 & 60 & 60 & 40 & 40 & 70 & 50 & 60 & 75 & 40 & 80 & 40 \\
40 & 10 & 60 & 0 & 80 & 50 & 10 & 60 & 70 & 90 & 50 & 70 & 60 & 40 & 80 & 50 & 10 & 60 & 70 & 90 \\
30 & 70 & 55 & 90 & 0 & 60 & 50 & 10 & 50 & 60 & 70 & 60 & 80 & 90 & 95 & 90 & 80 & 90 & 60 & 70 \\
80 & 60 & 50 & 60 & 80 & 0 & 70 & 40 & 35 & 50 & 40 & 40 & 70 & 40 & 70 & 40 & 35 & 50 & 40 & 40 \\
50 & 50 & 60 & 60 & 70 & 40 & 0 & 80 & 70 & 50 & 60 & 50 & 90 & 65 & 80 & 70 & 50 & 60 & 50 & 90 \\
60 & 60 & 45 & 55 & 80 & 30 & 90 & 0 & 40 & 60 & 60 & 40 & 70 & 80 & 90 & 95 & 90 & 80 & 90 & 30 \\
10 & 40 & 70 & 40 & 60 & 50 & 40 & 80 & 0 & 65 & 70 & 40 & 60 & 80 & 90 & 60 & 40 & 70 & 50 & 60 \\
60 & 60 & 50 & 90 & 50 & 60 & 50 & 60 & 80 & 0 & 60 & 10 & 60 & 70 & 40 & 50 & 90 & 40 & 75 & 10 \\
45 & 80 & 90 & 95 & 90 & 80 & 90 & 60 & 60 & 90 & 0 & 70 & 80 & 90 & 95 & 90 & 80 & 90 & 40 & 95 \\
40 & 70 & 60 & 80 & 50 & 60 & 40 & 70 & 40 & 95 & 40 & 0 & 50 & 60 & 60 & 40 & 50 & 60 & 80 & 50 \\
95 & 80 & 95 & 90 & 90 & 70 & 90 & 80 & 80 & 90 & 85 & 95 & 0 & 50 & 60 & 45 & 55 & 80 & 30 & 90 \\
80 & 90 & 95 & 90 & 80 & 90 & 60 & 80 & 50 & 60 & 50 & 60 & 40 & 0 & 80 & 90 & 95 & 90 & 80 & 90 \\
50 & 70 & 80 & 60 & 40 & 70 & 60 & 85 & 70 & 60 & 70 & 30 & 60 & 50 & 0 & 30 & 10 & 40 & 10 & 40 \\
60 & 10 & 95 & 50 & 80 & 90 & 95 & 90 & 80 & 90 & 50 & 40 & 30 & 10 & 50 & 0 & 40 & 10 & 50 & 10 \\
30 & 80 & 90 & 95 & 90 & 80 & 90 & 20 & 50 & 60 & 80 & 90 & 95 & 90 & 80 & 90 & 0 & 50 & 60 & 20 \\
50 & 10 & 45 & 95 & 40 & 30 & 60 & 80 & 90 & 95 & 90 & 80 & 90 & 50 & 40 & 70 & 50 & 0 & 30 & 50 \\
50 & 70 & 80 & 60 & 40 & 70 & 60 & 85 & 70 & 60 & 70 & 30 & 90 & 40 & 50 & 30 & 60 & 50 & 0 & 30 \\
70 & 50 & 60 & 80 & 50 & 90 & 70 & 60 & 90 & 55 & 80 & 80 & 70 & 80 & 90 & 95 & 90 & 80 & 90 & 0
\end{bmatrix}
$$

$$T_7 = \begin{bmatrix}
0 & 80 & 85 & 90 & 90 & 85 & 95 & 60 & 80 & 85 & 90 & 90 & 85 & 95 & 30 & 60 & 80 & 50 & 10 & 70 \\
75 & 0 & 80 & 85 & 90 & 90 & 85 & 95 & 50 & 50 & 40 & 50 & 60 & 50 & 50 & 70 & 40 & 60 & 10 & 90 \\
60 & 70 & 0 & 50 & 60 & 75 & 40 & 80 & 40 & 60 & 60 & 40 & 40 & 70 & 50 & 60 & 75 & 40 & 80 & 40 \\
40 & 80 & 60 & 0 & 80 & 85 & 95 & 80 & 90 & 85 & 90 & 85 & 90 & 90 & 85 & 95 & 10 & 60 & 70 & 90 \\
30 & 70 & 55 & 90 & 0 & 80 & 85 & 90 & 90 & 85 & 95 & 60 & 80 & 90 & 95 & 90 & 80 & 90 & 60 & 70 \\
80 & 60 & 50 & 60 & 80 & 0 & 70 & 80 & 85 & 90 & 90 & 85 & 95 & 40 & 70 & 40 & 35 & 50 & 40 & 40 \\
50 & 50 & 60 & 60 & 70 & 40 & 0 & 80 & 70 & 50 & 60 & 50 & 90 & 65 & 80 & 70 & 50 & 60 & 50 & 90 \\
60 & 60 & 45 & 55 & 80 & 30 & 90 & 0 & 80 & 85 & 90 & 90 & 85 & 95 & 90 & 95 & 90 & 80 & 90 & 30 \\
85 & 95 & 80 & 90 & 85 & 90 & 40 & 80 & 0 & 65 & 70 & 40 & 60 & 80 & 90 & 60 & 40 & 70 & 50 & 60 \\
60 & 60 & 50 & 90 & 50 & 60 & 50 & 60 & 80 & 0 & 60 & 10 & 60 & 70 & 40 & 50 & 90 & 40 & 75 & 10 \\
45 & 80 & 90 & 95 & 90 & 80 & 90 & 60 & 60 & 90 & 0 & 70 & 80 & 90 & 95 & 90 & 80 & 90 & 40 & 95 \\
40 & 70 & 60 & 80 & 50 & 60 & 40 & 70 & 40 & 95 & 40 & 0 & 50 & 60 & 85 & 95 & 80 & 90 & 85 & 90 \\
95 & 80 & 95 & 90 & 90 & 70 & 90 & 80 & 80 & 90 & 85 & 95 & 0 & 50 & 85 & 95 & 80 & 90 & 85 & 90 \\
80 & 90 & 95 & 90 & 80 & 90 & 60 & 80 & 50 & 60 & 50 & 60 & 40 & 0 & 80 & 90 & 95 & 90 & 80 & 90 \\
50 & 70 & 80 & 80 & 85 & 90 & 90 & 85 & 95 & 60 & 70 & 30 & 60 & 50 & 0 & 85 & 95 & 90 & 85 & 90 \\
60 & 10 & 95 & 50 & 80 & 90 & 95 & 90 & 80 & 90 & 50 & 40 & 30 & 10 & 50 & 0 & 40 & 10 & 50 & 10 \\
30 & 80 & 90 & 95 & 90 & 80 & 90 & 20 & 50 & 60 & 80 & 90 & 95 & 90 & 80 & 90 & 0 & 50 & 60 & 20 \\
85 & 95 & 80 & 90 & 85 & 90 & 60 & 80 & 90 & 95 & 90 & 80 & 90 & 50 & 40 & 70 & 50 & 0 & 30 & 50 \\
50 & 70 & 80 & 60 & 40 & 70 & 60 & 85 & 70 & 60 & 70 & 30 & 85 & 95 & 80 & 90 & 85 & 90 & 0 & 30 \\
70 & 50 & 60 & 80 & 50 & 90 & 70 & 60 & 90 & 55 & 80 & 80 & 70 & 80 & 90 & 95 & 90 & 80 & 90 & 0
\end{bmatrix}
\quad
T_8 = \begin{bmatrix}
0 & 80 & 85 & 90 & 90 & 85 & 95 & 60 & 80 & 85 & 90 & 90 & 85 & 95 & 95 & 85 & 80 & 90 & 95 & 70 \\
75 & 0 & 80 & 85 & 90 & 90 & 85 & 95 & 90 & 85 & 95 & 80 & 90 & 50 & 90 & 85 & 95 & 80 & 90 & 90 \\
60 & 70 & 0 & 90 & 85 & 95 & 80 & 90 & 90 & 85 & 95 & 80 & 90 & 80 & 40 & 90 & 85 & 95 & 80 & 90 \\
90 & 80 & 95 & 0 & 80 & 85 & 95 & 80 & 90 & 85 & 90 & 85 & 90 & 90 & 85 & 95 & 10 & 60 & 70 & 90 \\
30 & 70 & 55 & 90 & 0 & 80 & 85 & 90 & 90 & 85 & 95 & 60 & 80 & 90 & 95 & 90 & 80 & 90 & 60 & 70 \\
90 & 85 & 95 & 80 & 90 & 0 & 90 & 80 & 85 & 90 & 90 & 85 & 95 & 80 & 70 & 90 & 85 & 95 & 80 & 90 \\
50 & 90 & 85 & 95 & 80 & 90 & 0 & 80 & 00 & 50 & 60 & 50 & 90 & 65 & 80 & 70 & 50 & 60 & 50 & 90 \\
90 & 85 & 95 & 80 & 90 & 30 & 90 & 0 & 80 & 85 & 90 & 90 & 85 & 95 & 90 & 95 & 90 & 80 & 90 & 30 \\
85 & 95 & 80 & 90 & 85 & 90 & 40 & 80 & 0 & 90 & 85 & 95 & 80 & 90 & 90 & 90 & 85 & 95 & 80 & 90 \\
80 & 60 & 50 & 95 & 85 & 80 & 90 & 95 & 80 & 0 & 60 & 90 & 85 & 95 & 80 & 90 & 90 & 40 & 75 & 80 \\
95 & 80 & 90 & 95 & 90 & 80 & 90 & 60 & 60 & 90 & 0 & 70 & 80 & 90 & 95 & 90 & 80 & 90 & 90 & 95 \\
80 & 70 & 60 & 80 & 90 & 85 & 95 & 80 & 90 & 95 & 80 & 0 & 50 & 60 & 85 & 95 & 80 & 90 & 85 & 90 \\
95 & 80 & 95 & 90 & 90 & 70 & 90 & 80 & 80 & 90 & 85 & 95 & 0 & 50 & 85 & 95 & 80 & 90 & 95 & 90 \\
80 & 90 & 95 & 90 & 80 & 90 & 60 & 80 & 90 & 85 & 95 & 80 & 90 & 0 & 80 & 90 & 95 & 90 & 80 & 90 \\
50 & 70 & 80 & 80 & 85 & 90 & 90 & 85 & 95 & 60 & 70 & 30 & 60 & 50 & 0 & 85 & 95 & 90 & 85 & 90 \\
60 & 10 & 95 & 50 & 80 & 90 & 95 & 90 & 80 & 90 & 90 & 85 & 95 & 80 & 90 & 0 & 90 & 85 & 95 & 80 \\
30 & 80 & 90 & 95 & 90 & 80 & 95 & 85 & 80 & 90 & 95 & 90 & 95 & 90 & 80 & 90 & 0 & 50 & 60 & 20 \\
85 & 95 & 80 & 90 & 85 & 90 & 60 & 80 & 90 & 95 & 90 & 80 & 90 & 85 & 95 & 80 & 90 & 0 & 80 & 50 \\
90 & 70 & 80 & 90 & 85 & 95 & 80 & 90 & 85 & 70 & 60 & 70 & 85 & 95 & 80 & 90 & 85 & 90 & 0 & 30 \\
90 & 85 & 95 & 80 & 90 & 90 & 90 & 85 & 95 & 80 & 90 & 80 & 70 & 80 & 90 & 95 & 90 & 80 & 90 & 0
\end{bmatrix}$$

$$T_9 = \begin{bmatrix}
0 & 90 & 95 & 90 & 90 & 95 & 95 & 90 & 100 & 95 & 90 & 90 & 95 & 95 & 95 & 85 & 90 & 90 & 95 & 70 \\
95 & 0 & 90 & 85 & 90 & 100 & 85 & 95 & 90 & 95 & 95 & 80 & 90 & 90 & 90 & 85 & 95 & 80 & 90 & 90 \\
95 & 90 & 0 & 90 & 95 & 95 & 90 & 90 & 90 & 95 & 95 & 80 & 100 & 90 & 95 & 90 & 85 & 95 & 90 & 95 \\
90 & 95 & 95 & 0 & 95 & 95 & 95 & 95 & 95 & 95 & 90 & 85 & 90 & 100 & 85 & 95 & 90 & 90 & 95 & 90 \\
95 & 90 & 85 & 95 & 0 & 90 & 95 & 90 & 90 & 95 & 95 & 100 & 90 & 90 & 95 & 90 & 95 & 90 & 90 & 70 \\
90 & 95 & 95 & 80 & 95 & 0 & 95 & 90 & 95 & 90 & 90 & 85 & 95 & 90 & 90 & 90 & 85 & 95 & 100 & 95 \\
95 & 100 & 85 & 95 & 90 & 90 & 0 & 90 & 90 & 80 & 90 & 90 & 90 & 95 & 80 & 70 & 90 & 90 & 95 & 90 \\
95 & 85 & 95 & 90 & 90 & 80 & 90 & 0 & 100 & 85 & 90 & 90 & 85 & 95 & 90 & 95 & 90 & 80 & 90 & 85 \\
95 & 95 & 100 & 90 & 95 & 90 & 85 & 90 & 0 & 90 & 85 & 95 & 95 & 90 & 95 & 90 & 95 & 95 & 80 & 90 \\
90 & 85 & 95 & 95 & 85 & 90 & 100 & 95 & 95 & 0 & 90 & 90 & 95 & 95 & 90 & 95 & 95 & 90 & 75 & 80 \\
95 & 80 & 90 & 95 & 100 & 80 & 95 & 90 & 95 & 90 & 0 & 95 & 80 & 90 & 95 & 90 & 80 & 90 & 90 & 95 \\
80 & 100 & 95 & 80 & 90 & 95 & 95 & 80 & 90 & 95 & 90 & 0 & 90 & 95 & 95 & 95 & 90 & 95 & 85 & 90 \\
95 & 80 & 95 & 90 & 90 & 70 & 90 & 80 & 90 & 90 & 85 & 95 & 0 & 100 & 95 & 95 & 90 & 90 & 95 & 95 \\
100 & 90 & 95 & 90 & 80 & 90 & 80 & 90 & 90 & 85 & 95 & 80 & 90 & 0 & 90 & 95 & 95 & 95 & 90 & 90 \\
95 & 90 & 95 & 90 & 85 & 95 & 90 & 85 & 95 & 90 & 70 & 90 & 90 & 100 & 0 & 85 & 95 & 90 & 95 & 90 \\
85 & 95 & 95 & 90 & 95 & 90 & 95 & 90 & 90 & 100 & 90 & 85 & 95 & 80 & 90 & 0 & 90 & 95 & 95 & 80 \\
90 & 95 & 90 & 95 & 90 & 85 & 95 & 85 & 95 & 90 & 95 & 100 & 95 & 90 & 90 & 95 & 0 & 90 & 90 & 70 \\
85 & 95 & 95 & 90 & 95 & 90 & 95 & 100 & 90 & 95 & 90 & 85 & 90 & 95 & 95 & 90 & 90 & 0 & 90 & 95 \\
95 & 90 & 100 & 90 & 95 & 95 & 80 & 90 & 95 & 90 & 95 & 90 & 95 & 95 & 85 & 90 & 95 & 90 & 0 & 90 \\
90 & 95 & 95 & 90 & 90 & 100 & 90 & 95 & 95 & 95 & 90 & 90 & 95 & 80 & 90 & 95 & 90 & 85 & 90 & 0
\end{bmatrix}$$

$$T_{10}=\begin{bmatrix}
0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0 & 100 \\
100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 100 & 0
\end{bmatrix}$$

Bibliography

1 [ABJ02] Y. P. Aneja, S. Bandyopadhyay and A. Jaekel, "An efficient protection scheme for WDM Networks using dynamic lightpath allocation", University of Windsor, Aug. 2002.

2 [AMM00] S. Arakawa, M. Murata, and H. Miyahara, "Design methods of multiplayer survivability in IP over WDM networks", in Proc. OptiComm. TX. Oct. 2000.

3 [Bra90] C. Brackett, "Dense Wavelength Division Multiplexing Networks: Principles and applications", IEEE J. of Selected Areas in Communications, pp 373–380, Aug. 1990.

4 [Bra93] C. A. Brackett, et al., "A Scalable Multiwavelength Multihop Optical Network: A Proposal for Research on All-Optical Networks", Journal of Lightwave Technology, vol. 11, no. 5/6, pp. 736-753, May/June 1993.

5 [CB98] O. Crochat, and Jean-Y.L. Boudec. "Design Protection for WDM Optical Networks", IEEE J. Select Areas Commun. Vol. 16, no. 7, Sept. 1998, pp. 1158-1165.

6 [CGK92] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WAN's", IEEE Trans. Commun., vol. 40, pp. 1171–1182, July 1992.

7 [CGLM+00] J. S. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux, and D. Su, "A Functional Classification of Routing and Wavelength Assignment Schemes in DWDM Networks: Static Case", Published at Proceedings of the 7th International Conference on Optical Communications and Networks, OPNET, Page 1109-1115. Paris, France, January 2000.

8 [Chl93] I. Chlamtac et al., "Lightnets: topologies for high-speed optical networks", IEEE/OSA Journal of Lightwave Technology, vol. 11, pp. 951-961, May/June, 1993.

9 [Coc95] P. Cochrane, "Optical Network Technology – Foreword", New York: Chapman & Hall, 1995.

10 [CPTD98] B. V. Caenegem, W. V. Parys, F. De Turck, and P. M. Demeester, "Dimensioning of survivable WDM networks," IEEE J. Select. Areas Commun., vol. 16, Page 1146 – 1157, Sep. 1998.

11 [DDHH+99] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang, "Optical network design and restoration", Bell Labs tech. J., Page 58 – 83. Jan. – Mar. 1999.

12 [DG00] J. Doucette and W.D. Grover, "Influence of modularity and economy-of-scale effects on design of mesh-restoration DWDM networks", IEEE J. Select. Areas Commun., vol. 18, Page 1912 – 1923, Oct. 2000.

13 [EHS00] G. Ellinas, A. G. Hailemariam and T. E. Stern. "Protection Cycles in Mesh WDM Networks", IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, pp. 1924-1937, October 2000.

14 [Fum99] A. Fumagalli et al., "Survivable networks based on optimal routing and WDM self healing rings", Proc. of IEEE INFOCOM '99. Page 726 – 733, March 1999.

15 [Ger96a] O. Gerstel et al., "Dynamic channel assignment for WDM optical Networks with little or no wavelength conversion", Proc. 34th Annual Allerton Conf., Page 32 – 43, Monticello, IL, Oct. 1996.

16 [Ger96b] O. Gerstelet al., "Dynamic wavelength allocation in WDM ring networks", IBM Research Report RC 20462, May 1996.

17 [Ger98] O. Gerstel, "Opportunities for optical protection and restoration", in Optical Fiber Communication Conf., vol. 2, Page 269 – 270, San Jose, CA. Feb. 1998.

18 [GR00] O. Gerstel and R. Ramaswami, "Optical Layer Survivability: A Services Perspective", IEEE Commun. Mag., pp. 104-113, Mar. 2000.

19 [Gre92] P. E. Green, "Fiber Optical Networks", Prentice Hall, Englewood Cliffs, NJ 1992.

20 [Gre96] P. E. Green, "Optical Networking Update", IEEE Journal on Selected Areas in Communications, vol. 14, no. 5, pp. 764-779, June 1996.

21 [GRS97] O. Gerstel, R. Ramaswami and G. Sasaki, "Fault Tolerant Multi-wavelength Optical Rings with Limited Wavelength Conversion," in Proc., IEEE INFOCOM'97, (K0be, Japan), pp. 508-516, April 1997.

22 [Gua03] Hong Guan, "Optimal Logical Topology Design with WDM Path Protection", Master Thesis, 2003.

24 [HABJ+02] Anwar Haque, Yash P. Aneja, Subir Bandyopadhyay, Arunita Jaekel, and Abhijit Sengupta, "Some studies on the logical topology design of large multi-hop optical networks", Optical Networks Magazine, Vol. 3, No. 4, July/August 2002.

25 [IG00] R. R. Irashko, and W. D. Grover. "A highly Efficient Path-Restoration Protocol for Management of Optical Network Transport Integrity", IEEE J. Select. Areas Commun., Vol. 18, No. 5, pp. 779-794, May 2000.

26 [IL01] By ILOG, "ILOG CPLEX 7.5, Getting Started", Nov. 2001.

27 [Joe93] C. Joergensen et al., "4 Gb/s Optical Wavelength Conversion Using Semiconductor Optical Amplifiers", IEEE Photo. Tech. Lett., Vol. 5, No. 6, pp. 657-670, Jun. 1993

28 [LA91] J. P. Labourdette, A. S. Acampora, "Logically Rearrangeable Multihop Lightwave Networks", IEEE Transactions on Communications, Vol.39, No. 8, Aug. 1991.

29 [LDS98] E. Limal, S.L. Danielsen and K.E. Stubkjaer, "Capacity utilization in resilient wavelength-routed optical networks using link restoration", Proc., OFC '998, San Jose, CA, vol. 2, Page 297 – 198, Feb. 1998.

30 [LTS01] Y. Liu, D. Tipper, and P. Siripongwuikorn, "Approximating Optimal Spare Capacity Allocation by Successive Survivable Routing", INFOCOM 2001. Proceedings IEEE, vol. 1, 2001.

31 [Mar93] M. A. Marsan et al. "Topologies for Wavelength Routing All-Optical Networks", IEEE/ACM Tran. Networking, Vol. 1, Page 534 – 546. Oct. 1993.

32 [MN01] Eytan Modiano and Aradhana Narula, "Survivable routing of Logical Topologies in WDM Networks", INFOCOM 2001 Proceedings, IEEE, Vol. 1, Anchorage, pp. 348-357, 2001.

33 [MPPM02] G. Maier, A. Pattavina, S. D. Patre, and M. Martinelli, "Optical Network Survivability: Protection Techniques in the WDM Layer", Photonic Network Communication, 4:3/4, 251-269, 2002.

34 [MS98] Y. Miyao and H. Saito, "Optimal design and evaluation of survivable WDM transport networks," IEEE J. Select. Areas Commun., vol. 16, Page 1190 – 1198, Sep. 1998.

35 [MSYZ95] M. J. Mahony, D. Simeonidu, A. Yu and J. Zhou, " The Design of the European Optical Network", IEEE/OSA Journal of Lightwave Technology, vol. 3(5), pp. 817-828, May 1995.

36 [MT01] E. Modiano, A. Narula-Tam, "Survivable routing of logical topologies in WDM Networks", INFOCOM 2001. Proceedings. IEEE, vol. 1, Page 348 – 357, 2001.

37 [Muk92a] B. Mukherjee, "WDM-Based Local Lightwave Networks Part I: Single-Hop Systems", IEEE Network, pp. 12-27, May 1992.

38 [Muk92b] B. Mukherjee, "WDM-Based Local Lightwave Networks Part II: Multihop Systems", IEEE Network, pp. 20-32, July 1992.

39 [Muk94] B. Mukherjee et al., "Some principles for designing a wide-area optical network", Proc. IEEE INFOCOM'94, Toronto, Canada. Jun. 1994.

40 [Muk97] B. Mukherjee, "Optical Communication Networks", McGraw-Hill, 1997, ISBN 0-07-044435-8.

41 [OB01] Asuman E. Ozdaglar and Dimitri P. Bertsekas, "Routing and Wavelength Assignment in Optical Networks", MIT LIDS REPORT P-2535, December 2001.

42 [Ram93] R. Ramaswami, "Multiwavelength lightwave networks for computer communication," IEEE Commun., Page 78 – 88. Feb. 1993.

43 [Ram95] R. Ramaswami et al., "Design of logical topologies for wavelength routed all-optical networks", Proc. IEEE INFOCOM'95, Page 1316 - 1325 (Boston, Apr. 1995).

44 [RL00] I. Rubin and J. Ling, "Failure Protection Methods for Optical Meshed-Ring Communication, Networks", IEEE J. Select, Areas Commun. Vol. 18, No. 10, Page 1950 – 1960, October 2000.

45 [RM98] S. Ramaswami and B. Mukherjee, "Fixed-alternate routing and wavelength conversion in wavelength routed optical networks", in Proc., IEEE Globecom'98, vol. 4, Sydney, NSW, Australia, Nov. 1998, pp. 2295-2302.

46 [RM99a] S. Ramaswami and B. Mukherjee, "Survivable WDM mesh networks Part II – Restoration", ICC'99. 1999 IEEE International Conference on Communications, Vol. 3. pp. 2023-2030, 1999.

47 [RM99b] S. Ramaswami and B. Mukherjee, "Survivable WDM Mesh Networks, Part I-Protection", Proc. of IEEE INFOCOM'99, pp. 744-751, March, 1999.

48 [RS95] R. Ramaswami and K. N. Sivarajan, "Routing and Wavelength Assignment in All-Optical Networks", IEEE/ACM Trans. on Networking, vol. 3, no. 5, pp. 489-500, 1995.

49 [RS96] R. Ramaswami and K. Sivarajan, "Design of logical topologies for wavelength routed all-optical networks", IEEE J. Select Areas commun., Vol. 14, No. 5. Page 840 – 851, June 1996.

50 [RS98] R. Ramaswami, K. N. Sivarajan, "Optical Networks: A Practical Perspective", Morgan Kaufmann Publishers, 1998.

51 [SA00] G. Sahin and M. Azizoglu, "Optical layer survivability: Single service-class case", in Proc. OptiComm 2000, TX. Oct. 2000.

52 [SB97] S. Subramaniam and R. Barry, "Wavelength assignment in fixed routing WDM networks", in Proc. IEEE ICC, Montreal, Canada, pp. 406-410, Nov. 1997.

53 [SB99] T.E. Stern, K. Bala, "Multiwavelength optical networks - a layered approach" (Addison Wesley Longman, Inc., 1999).

54 [SG00] D. Stamatelakis and W.D. Grover, "IP Layer Restoration and Network Planning Based on Virtual Protection Cycles", IEEE Journal on Selected Areas in Communs., vol.18, no. 10, Page 1938 – 1949, October 2000.

55 [Ski97] Steven S. Skiena, "The Algorithm Design Manual", Department of Computer Science, State University of New York, Stony Brook, NY 11794-4400, 1997, http://www2.toki.or.id/book/AlgDesignManual/book/book/book.htm.

56 [SRM02] L. Sahasrabudhe, S. Ramamurthy and B. Mukherjee, "Fault management in IP-over-WDM networks: WDM protection versus IP restoration" IEEE J. Select. Areas Commun., Vol. 20, No. 1, Page 21 - 33. Jan. 2002.

57 [SSS02] Murari Sridharan, Murti V. Salapaka, and Arun K. Somani, "A Practical Approach to Operating Survivable WDM Networks", IEEE 2002.

58 [SW99] J. Spath, H. WeiBschuh, "Investigation of protection strategies: problem complexity and specifics aspects for WDM networks", in: Proc. NOC'99, (Delft, Netherlands, June 1999), vol. 2, pp. 58-74.

59 [Wu92] T. Wu, "Fiber Network Service Survivability", Norwood, MA: Artech House, 1992.

60 [YJ01] S. Yan, J. Jue, "A heuristic routing algorithm for shared protection in connection – oriented networks", in: SPIE Proc. OptiComm 2001, vol. 4599, Denver, CO (August 2001), pp. 142-152.

61 [ZJM00] H. Zang, J.P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks", Optical Networks Magazine, Jan. 2000.

Vita Auctoris

Name: Min Hou
Place of Birth: P. R. China
Year of Birth: 1972

Education:

9 / 00 — 5 / 03 University of Windsor, Windsor, Ontario, Canada
M. Sc. in Computer Science

9 / 89 — 7 / 93 Harbin Engineering University, Harbin, China
B. E. in Computer Science

Experience:

6 / 97 — 4 / 00 **Computer Programmer**
Japan MKC. Computer Co., Ltd., Beijing, China

7 / 93 — 6 / 97 **Network Administrator**
Zhongchuan Architectural Design & Research Institute, Beijing, China