1-1-2005

Survivability through pre-configured protection in optical mesh networks

Quoc V. Phung
Edith Cowan University

SURVIVABILITY THROUGH PRE-CONFIGURED PROTECTION IN OPTICAL MESH NETWORKS

By
Quoc Viet Phung

This thesis is presented in fulfilment of the requirements for the degree of
Master of Engineering Science
at
SCHOOL OF ENGINEERING AND MATHEMATICS
EDITH COWAN UNIVERSITY

Supervisor
Dr. Daryoush Habibi

January 6, 2005

DECLARATION

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education.

(ii) contain any material previously published or written by another person except where due reference is made in the text; or

(iii) contain any defamatory material.

I also grant permission for the Library at Edith Cowan University to make duplicate copies of my thesis as required.

Signature: [redacted]

Date: 18/01 2005

Abstract

Network survivability is a very important issue, especially in optical networks that carry huge amount of traffic. Network failures which may be caused by human errors, malfunctional systems and natural disaster (eg: earthquakes and lightning storms), have occurred quite frequently and sometimes with unpredictable consequences. Survivability is defined as the ability of the network to maintain the continuity of services against the failures of network components. Pre-configuration and dynamic restoration are two schemes for network survivability. For each scheme, survivability algorithms can be applied at either Optical Channel sublayer (OCh) known as link-based, or, Optical Multiplex Section sublayer (OMS) known as path-based. The efficiency of survivability algorithms can be assessed through such criteria as capacity efficiency, restoration time and quality of service. Dynamic restoration is more efficient than pre-configuration in terms of capacity resource utilization, but restoration time is longer and 100% service recovery cannot be guaranteed because sufficient spare capacity may not be available at the time of failures. Similarly, path-based survivability offers a high-performance scheme for utilizing capacity resource, but restoration time is usually longer than link-based survivability.

This thesis focuses on survivability of the network at both physical and logical layers. For survivability at the physical layer, we propose a theoretical framework to verify if a topology is survivable and identify the weaknesses of the network in terms of its survivability. For survivability of the logical topology, this thesis investigates pre-configured protection against single link failures. This is an optimization problem that we refer to as the *Survivable Logical Topology Design* (SLTD). We proposed an integrated objective function that can control the balance between the network utilization and the congestion level in the network. Finally, since SLTD has been proven to be an NP-hard, an new heuristic approach is devised to resolve the trade-off between the optimality of solutions and the computational time. This approach attempt to combine the computational advantages of the approaches based on graph theory and the optimality of solutions of Integer Linear Programming (ILP) with the small number of decision variables and constraints.

Acknowledgments

I wish to express my deepest appreciation to my advisor Dr. Daryoush Habibi. I was fortunate to have him as my mentor. I appreciate his intellectual exchanges, valuable suggestions and critical reviews. His knowledge and experience have been most helpful in ensuring the quality of this work.

I would like to thank my parents from Vietnam for their unconditional love and support. I'm also very fortunate to stay with my aunty family and my bother's family, specially when I first came to study in Australia.

Finally, special thanks to many friends of mine for helping me during the time of my study and the completion of the thesis.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

The advancement of optical networking technologies has evolved Wavelength Division Multiplexing (WDM) transmission systems as a solution to the rapid increase in the bandwidth requirement of todays networks. This is because the optical networks are capable of providing reliable transport medium with low-bit errors, high bandwidth and scalability. Compared with the traditional copper cables, optical fibers offer much higher bandwidth and are less vulnerable to various kinds of electromagnetic interferences. Furthermore, the advent of Wavelength Division Multiplexing (WDM) and Dense WDM (DWDM) techniques allow a more effective utilization of the tremendous bandwidth of optical fibers. A single fiber can carry many channels, and the total capacity can be increased dramatically [1]. Scientists from Bell Laboratories have reported that optical fiber can theoretically support 100Tbs or 100 trillion binary digits per seconds.

Given those dominant advantages, optical networks have been deployed and used as a high speed transport server layer carrying aggregate traffic of predominant client layers such as Internet Protocols (IP), Asynchronous Transfer Modes (ATM) and SONET/SDH as shown in Figure 1.1, adopted from [2]. In other words, traffic requirements from these client layers can be converted from electronic domain to optical domain and be bundled before they are carried on specific optical channels, routed through the network to the destination where the traffic is converted back to electronic domain. This process is referred to as the *Logical Topology Design* (LTD) problem. The LTD problem often includes two sub-problems: the *topology subproblem* and the *Routing and Wavelength Assignment* (RWA) subproblems. The topology subproblem determines a logical topology to be mapped on the physical topology; each logical link is an aggregated traffic connection bundled from higher transport layers. The RWA subproblem establishes routes and assigns wavelengths to the required traffic connections.

Such optical routes are referred to as *lightpaths*

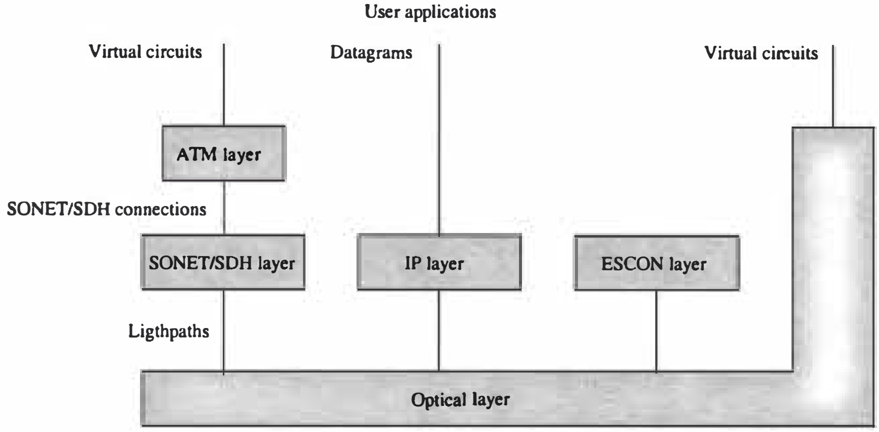


Figure 1.1: *The second-generation optical network layer that supports a variety of client layers*

In the first generation optical networks, lightpaths are point-to-point connections through a physical fiber and each fiber offers only one wavelength channel or one lightpath. In the second generation optical networks, the lightpaths are allowed to pass through several fiber links in various wavelength channels due to the advent of *Wavelength Division Multiplexing* (WDM) technology and optical elements such as *Optical Add-drop Multiplexers*(OADMs) and *Optical Cross-Connects* (OXCs). However, the cost of a lightpath, which includes the cost of optical equipments (OXCs, OADMs) and fibers, is still expensive, and hence, for optimization purposes, lightpaths are usually routed through the shortest paths or more generally through the least expensive paths. In general, the aim of the LTD is to optimize the operations of the network, both in network performance and resource utilization.

Being a backbone network, it is foreseen that there is a huge amount of traffic exchanged in the network at any one time, hence a failure of an optical component such as a fiber cut or a failure of a node may cause a very serious problem in terms of loss of data and profit. For instance, the Gartner research [3] attributes up to $500 million in business losses to network failures by the year 2004, or direct voice-calling revenue loss from failure of major trunk group is frequently quoted at $100,000/minutes or more. Network survivability, therefore, is becoming a critical and imperative problem in telecommunication networks today, particularly in optical networks. *Network survivability* by definition is the capability of the network to maintain the continuity of services against the failures. In this context, each working lightpath that is affected by a failure is switched to an alternative lightpath to maintain the service. The working lightpath is called the *primary path* and the alternative lightpath is called the *backup* *path.* The establishment of such backup paths may be either online or offline. With online provisioning, the backup paths are only determined after a failure occurs, and is referred to as *dynamic restoration.* The LTD with survivability, referred to as the *Survivable Logical Topology Design* (SLTD), in contrast, provisions the backup paths offline or in design phase. In other words, in SLTD scenario, the primary paths and backup paths of connections are simultaneously established. The aim of the SLTD problem is to provision traffic requirements over a given physical topology so that the continuity of the traffic is assured in case of failures. There are three key factors affecting to the performance of the SLTD problem: the physical topology, survivability schemes and the implementation of the survivability schemes.

1. *The physical topology.* Two most popular topologies successfully used in optical networks are mesh and ring structures. The main advantage of mesh arrangement is the ability to utilize the network resources more efficiently under the normal operation and hence allowing the minimization of network capacity requirements. In contrast, the ring structure may not be as efficient as mesh structure in terms of utilizing the network resources but it offers many interesting features. Firstly, a ring is a two-connected topology, so the algorithms for routing is not complicated, and thereby simplify the policy of survivability. Another useful feature of rings is the fast response to network failures. Since the restoration is automatically implemented at the hardware layer, it is very fast and reliable. In contrast, since mesh topology is more complex, the survivability algorithms are more complicated and require more computational time.

2. *Survivability schemes.* The performance of network survivability mainly depends on the survivability schemes used. For example, path protection scheme which refers to the restoration between end-nodes of a failed connection is efficient in network resources utilization while link protection scheme which performs the restoration between end-nodes of a failed link provides a fast restoration. In addition, with respect to the method by which wavelength channels are assigned to backup paths, dedicated protection schemes provide a reliable protection and faster restoration, compared to the shared protection schemes, but they require more spare capacity.

3. *The implementation of the survivability schemes.* The efficiency of a survivability scheme depends not only on the model itself but also on the approaches to implement the model. In other words, with the same survivability scheme, different implementations result in different solutions. The optimum solution can be achieved by using Integer Linear Programming (ILP) formulation, but this approach is intractable even with moderate scale networks. Approaches based on

graph theory can resolve the problem of computational time. These approaches, however, usually result in near-optimal solutions.

1.1 Aims of this thesis

In this thesis, we consider the logical topology design in the context of network survivability, known as *Survivable Logical Topology Design* (SLTD), with the objectives of improving network capacity utilization and reducing congestion levels. In addition, the survivability topology is investigated. Given these two key goals, our study focuses on the following objectives:

1.1.1 Survivability of physical topology

The protection requirements of traffic in networks are different from application to application. For example, a broadcast application may need no protection for the data, while a communication application in military may require a very high level of protection because of the importance of the transmitted data. The classification of protection levels can be done using different criteria. One criterion in [3] is based on the quality of service (QoS). In another criterion, protection levels can be classified according to the failures of network components, ie. network links and networks nodes. It is observed that the physical topology has to satisfy some specific conditions to be able to support each level of protection. For example, for protection against link failures, the physical topology must be able to offer at least one pair of link-disjoint paths between any two nodes in the network.

The first aim of this thesis is to dissect the problem of survivability at physical topology layer, point out the weaknesses and outline the open problems in this field. We attempt to build a theoretical framework for the classification of the survivable physical topologies that allows for different levels of protection required from logical topology design.

1.1.2 Survivability of logical topology

The SLTD attempts to route traffic connections through optical channels so that in case of a failure, the affected connections are switched to alternative paths to maintain the quality of services. In other words, for each traffic connection, the SLTD establishes two lightpaths from source node to destination node; one is used as the working path for normal operation and the other is used as the backup path to overcome network failures. The principles for routing and assigning capacity to these paths differ amongst survivability schemes such as path/link protection and dedicated/shared protection. For example, a backup path in path protection protects its working path between end-nodes of a connection whereas backup paths in link protection protects the working channels between end-nodes of the failed link. Hence, the performance of the SLTD is dependent on the network survivability schemes used. In addition, for each network survivability scheme, the performance of the SLTD is also dependent on the implementation of the schemes such as the ILP formulation and the graph theory approach. The ILP model offers an exact formulation but it is intractable even with moderate scale networks while algorithms based on graph theory usually offer fast computation but sometimes they may not find a solution even though it exists. In fact, SLTD is an optimization problem which has been proven to be NP-hard.

- As an optimization problem, the SLTD attempts to optimize specific objectives dependent on the requirements of network operations. One common objective used in the literature is to minimize the total number of wavelength channels used. Another objective is to minimize the congestion levels in the network. We have observed that, for schemes in which the congestion level is intended to be minimized, the number of utilized wavelengths in the network may be very high. This reduces the blocking probability for the next connections but the network resources may be quickly exhausted. Conversely, when the objective is to minimize the total number of wavelength channels used, we may get high the congestion levels in the network, ie. the total number of wavelength used *on some links* may reach to their limit although the total number of wavelength used *in the network* is low. Such links will block future connections which need to use those links. The objective here is to construct an objective function that can trade-off between these objectives, ie. 1) minimizing the total number of wavelengths used and 2) minimizing the congestion levels.

- The constructed objective function can be simply modeled using ILP formulation. However, the modeling of the objective function using the graph theory approach is not as simple. The second aim in the SLTD problem is to develop a suitable mathematical framework that allows to satisfy the requirement of the objective function through approaches based on graph theory.

- Having achieved the first two aims described above, we reach the main aim of this thesis in the context of SLTD, which is to investigate network survivability in logical topology design through existing protection schemes, including path/link protection and shared/dedicated protection. The performance of these schemes is compared in terms of network resource utilization and the network congestion level. All protection schemes are implemented using existing approaches, namely ILP formulation and graph theory approach.

- As known, there are a conflict between the optimality of solutions and the time complexity both in the ILP formulation and the graph theory approach. The final objective of this thesis is to propose a heuristic approach that can take the computational advantages of graph theory and the optimality of solutions of the ILP formulation by significantly reducing the number of integer variables and constraints.

1.2 Thesis contribution

As mentioned earlier, in this thesis, we consider the survivability of the optical network both for physical and logical topologies. The following items summarize the contributions of this thesis in each area.

- Considering that the survivability of the logical topology is heavily dependent on the survivability of the physical topology, establishing the physical survivability of the network is of utmost importance. However, existing techniques are not able to establish the physical survivability of a moderate size network in a reasonable amount of time. For instance, the cutset technique which shall be described later, is not applicable to a network which has 30 nodes. There is clearly a major problem here which is resolved by the contributions of this thesis in this area. We provide a novel theoretical framework, consisting of 2 new theorems and 3 new lemmas, all proven, for the assessment of the physical survivability of the network. Our framework can cope with very large size networks, even several thousand nodes, which were simply beyond the scope of any existing technique prior to this thesis.

- The theoretical framework for assessment of physical survivability of the network is implemented using a number of algorithms developed in this thesis. Two of these algorithms are modifications of existing spanning tree algorithms, and two of them are new algorithms that we have developed in this study.

- The implementation of our physical survivability framework can clearly identify the weaknesses of a network. For the first time, it is possible to establish the survivability of the network not only on the basis of link failures, but also with respect to node failures. No research has been able to achieve this in the past.

Furthermore, our solution gives a comprehensive diagnosis of the network and identifies the exact nodes and links which are the weaknesses of the network, making it unsurvivable.

- The next contribution of this thesis is in the area of survivable logical topology design (SLTD). One common SLTD objective is minimization of the total number of wavelength channels used. Another objective is to minimize the congestion levels in the network. These objectives are treated separately in the literature, that is, only one of them is targeted at a time. In this thesis, for the first time, we introduce an integrated objective function that can combine the two objectives in the optimization problem. Therefore the solution obtained minimizes the total number of channels utilized, at the same time as minimizing the congestion of the network.

- It is observed that the quality of the solution in the context of network survivability depends not only on the survivability scheme applied, but also to the specific implementation of the scheme. Although the integrated objective function, as mentioned in the last item, can be modeled using ILP formulation, it will limit the scalability of the solution, and as the network grows, the size of the problem will quickly get out of hand. Therefore, this research has developed a novel implementation of the integrated objective function based on the graph theory. This implementation has significant computational advantages over the classical ILP formulation, and can be applied to large size networks.

- It is a well known fact that in many optimization problems, there is a conflict between the optimality of solutions and the time complexity involved. The final contribution of this thesis is the development of a novel heuristic approach that can take the computational advantages of graph theory and the optimality of solutions of the ILP formulation. Our technique significantly reduces the number of integer variables and constraints, thus making the solution both optimal and time efficient.

1.3 Outline of the thesis

In **Chapter 2**, the background and the literature review for network survivability are presented. An overview of the principles and schemes in network survivability are presented and the mathematical framework for the study of the survivability is provided. In **Chapter 3**, we propose a theoretical framework for establishing the physical survivability of the network. In addition, an efficient approach for the verification of network survivability of arbitrary physical topology is proposed. This algorithm also point out the "handicap" of the topology regarding to survivability such as unconnected node, node-bridges or link-bridges. Next, in **Chapter 4**, the SLTD problem is investigated with different survivability schemes and different approaches. An new heuristic approach to balance the conflict of the optimality of the solution and time complexity is proposed in this chapter. Finally, the conclusions and suggestions for future extensions to this research are discussed in **Chapter 5**.

Chapter 2

An Overview of Network Survivability

The main objective of this research is to investigate the problem of network survivability in logical topology design, known as the *Survivable Logical Topology Design (SLTD)* problem. This mainly involves provisioning for traffic demands over a given physical optical network so that the continuity of the traffic is ensured in case of failures. Hence, two key issues that affect the performance of the solutions to the SLTD problem are: 1) the survivability schemes applied to networks; and 2) implementations of these survivability schemes. In the first issue, the restoration time and the restorability are two important metrics whose values allow designers to estimate the survivability of a network. On the other hand, performance of survivability scheme implementations is measured through the optimality of solutions and time complexity.

In this chapter, we investigate the SLTD with respect to the above issues and review some significant results from the literature. Firstly, four common survivability schemes in optical networks, namely link protection, path protection, dedicated protection and shared protection, are studied in depth. We summarize the strengths and weaknesses of these protection schemes. Secondly, since the SLTD is referred to as an optimization problem which is proven to be NP-hard, there is a trade-off between the optimality of the solutions and the time complexity in approaches to the problem. We investigate the problem through existing approaches in the literature, namely Integer Linear Programming (ILP) and graph theory approach. We analyse the strengths and weaknesses of these approaches and highlight some of their results.

We note that most of the research on SLTD has usually overlooked the physical topology of the network, or has assumed the physical topology is survivable. The problem with this assumption is that if the physical topology is not survivable, then seeking for survivability of the logical topology is obviously redundant. Thus, the physical topology survivability is investigated before we review and explore the SLTD problem.

The rest of this chapter is organized as follows: in Section 2.1, we give a general overview of optical network architecture. The survivability of physical topology is discussed in Section 2.2. In Section 2.3, we investigate survivability schemes in logical topology design. Finally, approaches to the SLTD problem are investigated and analysed in Section 2.4

2.1 Optical Network Architecture

In this section, the architecture of optical WDM network including network components and the network topology is investigated. Next, the concepts of optical networks through notations are introduced and highlighted.

2.1.1 Optical Network Components:

Optical networks, in general, include the following key components: optical line terminals, optical add/drop multiplexers, wavelength converters and optical cross connects. The position and functionality of these components in the network are presented as follows:

- *Optical Line Terminals (OLTs):*

  OLTs can be used at either end of a point-to-point link to multiplex/demultiplex wavelength channels from/to data of higher layers such as IP, ATM, and SONET/SDH. In addition, the OLTs also terminate an *optical supervisory channel* (OSC) [2, 4]. The OSC is carried on a separate wavelength, which differs from the wavelengths carrying the actual traffic. This is used to monitor the performance of amplifiers as well as other management functions.

- *Optical Add/Drop Multiplexers (ODAMs):*

  ODAMs are used to adjust the flow of traffic through optical networks and have three functions: 1) optical traffic can pass over the devices without any interruption or optoelectronic conversions; 2) optical traffic can be terminated or *dropped* from some specific wavelength channels and converted to electronic domain; and

3) data from electronic domain is converted to optical domain and *added* to wavelength channels.

- *Wavelength Converters (WCs):*

  WCs are used to convert data from one wavelength channel to another channel. Without wavelength conversion, data from a source node to a destination node has to be carried on the same wavelength channels. Wavelength conversion can be classified into three categorizes: *full wavelength conversion*, *limited wavelength conversion*, and *fixed wavelength conversion.* Full wavelength conversion allows any input wavelength channel to be converted to any wavelength channel at output; limited wavelength conversion implies that each input wavelength channel can be converted to the specific set of output wavelength channels; and fixed wavelength conversion is a special case of limited wavelength conversion in which each input wavelength channel is converted to exactly one output wavelength channel.

- *Optical Cross connects OXCs:*

  The advent of OXCs enhances the flexibility of optical networks. These devices are used to switch optical data on desired routes. An OXC provides the two following key functions in large networks [2]: 1) An OXC can be used to provision lightpaths in an automated manner, without having to resort to performing manual patch panel connections; and 2) most importantly, it can provide the protection capability against fiber cuts and equipment failures. Therefore, an optical network equipped with such devices is considered as a virtual circuit switching. The use of OXCs mainly is for reducing the traffic blocking.

2.1.2 Network Topologies:

The topology of a network denotes the connectivity of that network. A network can generally be modelled as a connected graph in which there exist at least one path between any two nodes in the network. Traffic requirements between two nodes are routed through these paths. In this part, we introduce two popular topologies in optical networks, namely rings and mesh topologies; and highlight the advantages and disadvantages of the two with regards to network survivability.

- *Ring topology:*

  Most WDM optical networks today are based on the ring topology, especially in metro or regional areas [5]. Ring topologies offers a pair of disjoint paths (both node-disjoint and link-disjoint) between any two nodes. One path is in clockwise direction and the other is in reversed direction. Thus, with respect to network survivability, rings are simple both in implementation and operation. However, protection schemes over rings often requires a high spare capacity for backup, 100% in theory, but even over 200% in reality [6].

A typical ring topology is a SONET Self-Healing Ring (SHR). When a component in the ring (a network node or link) fails, the affected traffic is rerouted along the opposite direction of the ring to recover connections. Two common kinds of SONET SHR networks are unidirectional SHR (USHR) and bidirectional SHR (BSHR) [1]. USHRs include two parallel optical rings, one serves as the primary ring and another as the protection ring. In normal operation, traffic is carried in one direction of the primary ring. The traffic is switched to the protection ring when failures occur. BSHRs, also called BLSRs (Bidirectional Line-Switched Rings), are divided into two architectures due to two or four fibers used for protection. A BLSR/2 contains two protection fiber in which half of its capacity on each ring is reserved for protection. The reserved capacity of one ring is used as protection primary capacity in another ring, and vice versa. A BLSR/4 contains four fibers rings: two rings are dedicated for the protection purpose and the other two fibers serve as primary rings.

- *Mesh topology:*

  Though the ring topology is the most popular physical topology today, WDM mesh topologies are becoming more important due to the advent of optical switches (OXCs). The survivability in the mesh topology is much more complex than ring topology because of the numerous options for routing, but it more flexible and scalable. Mesh topologies allow designers to employ many protection schemes and allow researchers to develop better protection schemes that can improve the network resource utilization or improve survivability performance.

2.1.3 Network Notation:

In this part, network terminologies, notations and definitions which are used in the coming sections and chapters are introduced.

- **Physical Topology.** A structure of a physical optical network is denoted as a graph $G(V, E)$ in which each optical node is a vertex in V, and each fiber link between any two nodes is an edges in E. Fiber links are usually assumed to be bidirectional, that is, traffic can be carried on either directions of a wavelength channel on fiber links, and hence graph G is usually an undirected graph. There is a weight associated with each edge which is usually the cost of a fiber link. In our study, since we only consider the number of wavelengths used on a link, the weight of all links is assigned by 1, that is, the cost of each edge in the graph is 1.

- **Wavelength channels.** The bandwidth of a fiber link between two nodes in the optical WDM networks is partitioned into many distinct channels, called *wavelength channels* or *optical channels.*

- **Lightpath.** An optical connection between two nodes in the network is known as a lightpath. Such connection can carry traffic data from one node to another without any conversion between electronic domain and optical domain. In the absence of the wavelength conversion, optical channels contained in a lightpath are assigned with the same wavelength. This is called *the wavelength continuity constraint.*

- **Logical Topology.** We model a logical topology to as a graph that has as same set of network nodes as the physical graph. Edges of the logical topology are lightpaths. In the SLTD problem, traffic connections require to be routed on lightpaths, and hence, the logical topology differs from one set of traffic connections to another. The objective of logical topology design, in our study, is to setup lightpaths for the given traffic connections. Since traffic requirements are usually directed connections from one node to another, logical topologies are directed graphs.

- **Nodal degree.** Nodal degree of a node is the number of physical fiber links connected to the node.

- **Link order.** A physical link in the network between node i and node j is denoted as $e_k \equiv e_{ij}$.

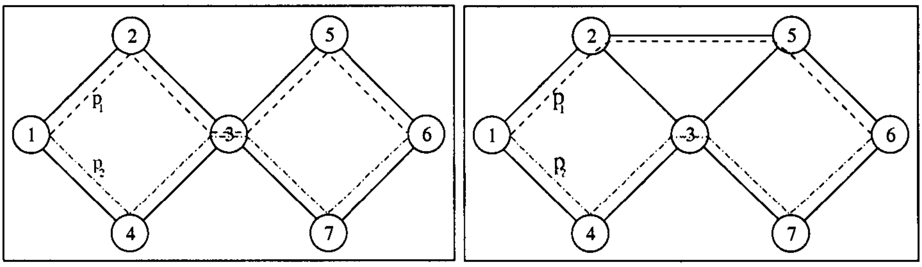
- **Traffic matrix.** The long-term/estimated traffic on the network is modeled as an $N \times N$ traffic matrix T, where N is the number of network nodes. The value of an element t^{sd}, $(s,d) \in \{1 \dots N\}$ in T denotes the number of connections required from the source node s to the destination node d.

2.2 The Survivability of Physical Topology

Physical layer is the infrastructure of physical resources on which the network is based on: buildings, right-of-ways, cable ducts, cables, underground vaults, and so on [3]. At the physical network design level, there are number of standard practices to enhance network survivability such as substances for protecting cables, the depth of burying cables and warning signs. These standards practices differ from one area to another to reduce susceptibility from human activities. Protection standards, however, only aim to reduce the probability of failures in the network. In addition, the cost of trenching for burying cables can be quite significant due to the depth of burial required and the nature of geographical regions. Restoration schemes are considered to improve network survivability and reduce the cost of burying cables. This is concerned with the physical topology of the network.

The concept of survivability at the physical layer is mainly based on graph theory. Hence, terminology from graph theory [7, 8] is adopted as follows:

- A physical topology is represented by a graph $G(V, E)$, where V is a set of network nodes and E is a set of network links.
- A *path* between any two nodes is a sequence of consecutive nodes and links from the original node to the target node. Note that a path only traverse over a node or a link at most once.
- A *cycle* is a closed path, that is, the origin and destination nodes of the cycle are the same.
- A graph is *connected* if there exists at least one path between any two nodes in the graph.
- A pair of two paths between two nodes is *link-disjoint* if the two paths do not share any link.
- A pair of two paths between two nodes is *node-disjoint* if the two paths do not share any node.



(a) *Two-connected graph*

(b) *Biconnected graph*

Figure 2.1: *Survivable Networks*

Wayne D. Grover in [3] has pointed out that a survivable topology must be based on a two-connected or preferably a biconnected graph. A graph is two-connected if it has at least one pair of link-disjoint paths between every two nodes in the graph. A graph is biconnected if it has at least one pair of node-disjoint paths between any two nodes in the graph. For example, Figure 2.1(a) shows a two-connected graph, but not a biconnected graph since any pair of two disjoint paths between nodes 1 and 6 is link-disjoint. However, when link $(2,5)$ is added to this graph as in Figure 2.1(b), the graph is biconnected. Graph connectivity properties, therefore, are important for transport networks in terms of survivability.

Both two-connected graphs and biconnected graphs are suitable for survivability against link failures, but biconnected graphs can also survive nodes failures. A survivable network, in general, must be in the form of either these types of graphs in order to support survivability schemes at logical topology layer. However, research in logical topology design has usually overlooked the physical topology survivability or has assumed that the given physical topology is survivable [9, 10]. This leads to a so-called "check-redundant" dilemma, that is, if the physical topology is not survivable, then seeking for survivability at logical topology layer is obviously redundant. In addition, while survivable networks can easily be recognized by human inspection in small networks, efficient algorithms for checking survivability in large scale networks are required to avoid human errors. An algorithm for finding biconnected components of a graph is introduced in [3]. This algorithms is based on a depth-first search (DFS) in a graph, followed by a backtracking phase. This algorithm, however, only results in a maximal biconnected component. In other words, a graph is biconnected if the number of nodes in the resulting biconnected component of this algorithm is equal to number of network nodes. In addition, the algorithm does not indicate the weaknesses of the topology such as node-bridges or link-bridges. Other algorithms based on the *cut-set* method can be used to verify network survivability at physical topology. However, such algorithms can only imply if a network is survivable, that is, the topology of the network can be either two-connected or biconnected. Details of the method will be discussed further in Chapter 3.

2.3 Network Survivability in Logical Topology Design

Logical Topology Design (LTD) in optical WDM networks often includes two subproblems, namely *topology design* and *Routing and Wavelength Assignment (RWA).* Topology design determines lightpaths for provisioning data from higher transport layers such as IP, ATM, and SONET/SDH. In our study, traffic from these client layers is assumed to be groomed into traffic bundles and provisioned through optical connections, and hence, topology design will not be considered in this thesis. We refer the LTD problem to as the second sub-problem, the RWA problem whose objective is to establish a route and assign a specific wavelength to the route for each traffic request.

The provisioned connections, however, are susceptible to network failures. In order to maintain the continuity of the services, connections affected by failures must be rerouted and switched to alternative routes. This capability amongst network performance is referred to as network survivability. The survivable provisioning for a traffic connection, therefore, requires as finding two paths between end-nodes of the connection. One path is employed to carry data in normal operation and is denoted as *primary path.* Another path is assigned as a *backup path* which carries data in case of a failure in the primary path. The relationship between the primary path and the backup path can vary depending on the levels of protection requirements. For example, if protection is required against link failures, a pair of primary and backup paths is link-disjoint whereas the path-pair need to be node-disjoint if protection requirement is to against both node failures and link failures.

In addition, network survivability schemes can also vary depending on the levels of protection requirements. The path protection scheme is employed to provision connections that do not require fast restoration whereas link protection is designed for this purpose. The dedicated protection scheme is used to assured 100% restorability while in the shared protection scheme, the restorability of connections can be controlled using various algorithms. Figure 2.2 shows a classification of these protection schemes.

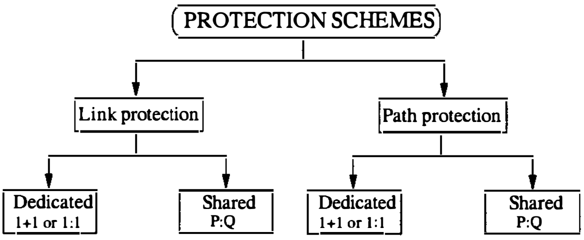


Figure 2.2: *The classification of protection schemes*

In the rest of this section, we introduce the operational mechanism of these survivability schemes and review some significant results from the literature.

2.3.1 Path protection versus link protection

The operation of path protection and link protection schemes is based on the performance of sublayers of the optical transport layer. According to the ITU-T Recommendation G.872 [4], the optical transport layer is mainly divided into four sublayers, namely the Optical Channel (OCh), the Optical Multiplex Section (OMS), the Optical Transmission Section (OTS), and the Physical Media [5] as shown in Figure 2.3.

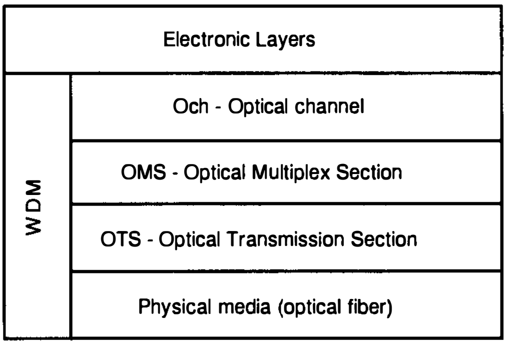

Electronic Layers
WDM
Och - Optical channel
OMS - Optical Multiplex Section
OTS - Optical Transmission Section
Physical media (optical fiber)

Figure 2.3: *Optical Transport network protocol architecture: sublayers of WDM layer*

1. *The Optical Channel sublayer:*

   This sublayer manages all the end-to-end networking functions including routing, wavelength assignment, fault recovery, and so on. Since the fault recovery is performed from one end-node to the other end-node of a connection, the protection schemes implemented in this sublayer are referred to as *path protection.*

2. *The Optical Multiplex Section sublayer:*

   This sublayer multiplex the WDM channels carried on a single fiber link. This sublayer mainly performs WDM multiplex monitoring such as checking the integrity of the multiplexing process and wavelength stability. Since the OMS is operated between end-nodes of a link, protection schemes in this sublayer is referred to as *link protection.*

3. *The Optical Transmission Section sublayer:*

   All the control operations of optical devices such as transponders, regenerations, and amplifiers are undertaken by this sublayer.

4. *The Physical Media:*

   This sublayer provides a point-to-point WDM transmission medium.

The full scheme of the optical transport layer is more complicated. However, since our study aims to investigate protection schemes against link failures, we will focus on path protection and link protection.

Path protection

Path protection is referred to as fault management between end-to-end of optical connections. When a link in the network fails, the source nodes and the destination nodes of affected routes are informed via messages from the nodes adjacent to the failed link. Then, traffic in the primary routes are switched to alternative routes to maintain the continuity of the connections. Figure 2.4 shows an example of the main operational mechanism of path protection. Under normal operation of the network, data for connections $(1-4)$ and $(6-4)$ are carried on two primary routes $p_1 = (1 \rightarrow 6 \rightarrow 5 \dashrightarrow 4)$ and $p_2 = (6 \rightarrow 5 \dashrightarrow 4)$. The two paths traverse through link $(6-5)$ that fails in this example. Path protection switches data on these primary path to their alternative paths which are remarked as $r_1 = (1 \rightarrow 2 \dashrightarrow 3 \rightarrow 4)$ and $r_2 = (6 \dashrightarrow 3 \dashrightarrow 4)$. Note that, in this example, pairs of two paths (p_1, r_1) and (p_2, r_2) are node-disjoint.

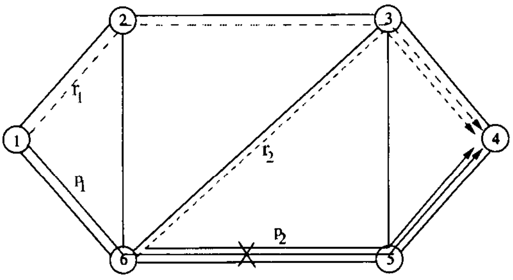


Figure 2.4: *Path protection*

Backup paths can be provisioned either online or offline. With online provisioning, backup paths are only determined after a failure occurs. Since available network capacity is non-deterministic, online provisioning does not ensure 100% quality of service. Connections may be blocked because the network capacity may be exhausted at time of failure. Furthermore, online provisioning requires fast algorithms. Offline provisioning, on the other hand, sets up a primary path and a backup path simultaneously at the design phase. In this case, the primary path and the backup path must be disjoint (link-disjoint or node-disjoint) against link failures. Since backup paths in offline provisioning are dedicated and reserved for their primary paths, this scheme of protection requires more spare capacity compared with online provisioning, but it assures 100% of restoration. In addition, offline provisioning can offer fast restoration because of the availability of backup paths. On the other hand, with offline provisioning, a set of primary and backup paths of connections need to be provisioned simultaneously, leading to an optimization problem which has been proven to be NP-hard.

Link protection

Link protection refers to protection schemes applied at OMS sublayer on optical WDM transport layer. That is, all primary routes on a failed link are switched to their backup paths around end-nodes of the failed link as shown in Figure 2.5. When link $(5-6)$ fails, two channels of primary paths p_1 and p_2 are switched to backup paths $r_1 = (6 \rightarrow 2 \rightarrow 3 \rightarrow 5)$ and $r_2 = (6 \rightarrow 3 \rightarrow 2)$ to maintain the continuity of these connections. These backup paths are between end-nodes of link $(5-6)$, not between end-nodes of connections.

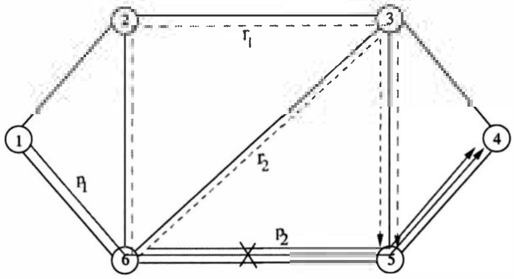


Figure 2.5: *Link protection*

In WDM networks, a fiber link contains many distinct wavelength channels, and hence a failure of a single link leads to the failure of all channels within the link. Different primary channels on a failed link can have different backup paths. Similar to path protection, backup paths of primary channels may be provisioned at either online or offline. The advantages and disadvantages of online and offline provisioning in link protection are as same as those described for path protection.

Table 2.1: *Path protection versus link protection*

Path Protection	Link Protection
• End-to-end detouring • Better resource utilization	• Local detouring • Faster restoration

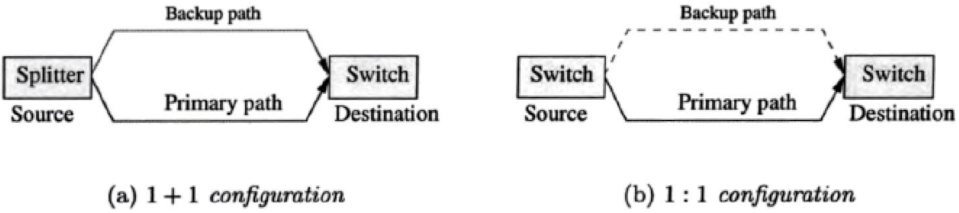
In SLTD problem, traffic connections are known in advance, and hence the primary and backup paths of the traffic connections for both path and link protection are provisioned as offline provisioning. Existing approaches to these protection schemes (path/link protection) are discussed in Section 2.4. Table 2.1 summarizes the performance of these protection schemes achieved from the literature [1, 5, 11]. Path protection is an end-to-end detouring in which backup paths are discovered between end-nodes of connections, not between end-nodes of fiber links. In contrast, link protection is recognized as local detouring. On the other hand, link protection offers a faster restoration, compared to path protection, whereas path protection offers a better capacity utilization than link protection [10, 1].

2.3.2 Dedicated protection versus shared protection

Protection schemes for link/path protection can commonly be divided into dedicated protection and shared protection according to the mechanism of capacity allocation on backup paths. When each backup path is a dedicated optical connection, the protection scheme is called a *dedicated protection.* On the other hand, if the backup paths use some wavelength channels in common, the schemes is referred to as *shared protection.* Thus, protection schemes in general, can be classified into four categories: dedicated path protection, dedicated link protection, shared path protection and shared link protection. Except explicitly stated otherwise, we shall refer to shared protection as either shared path protection and shared link protection; and similarly, dedicated protection is referred to as either dedicated path protection or dedicated link protection.

Dedicated protection

Dedicated protection in both path and link protection has two configuration, namely $1 + 1$ (one plus one) configuration and $1 : 1$ (one by one) configuration as shown in Figure 2.6.



(a) $1+1$ *configuration* (b) $1:1$ *configuration*

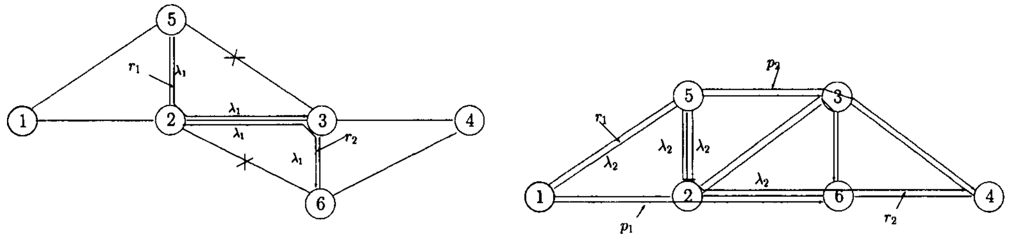
Figure 2.6: *Dedicated protection schemes*

In the $1 + 1$ configuration as shown in Figure 2.6(a), the signal of a connection is simultaneously transmitted on a primary path and a backup path, and hence a splitter is used at the source node to split the transmitted signal between these paths. The receiver at the destination node compares the two signals and selects the better one. When a failure occurs in the working path, the receiver automatically switches to the remaining path. On the other hand, in the $1 : 1$ configuration as shown in Figure 2.6(b), the signal of a connection is only transmitted on the primary path, and the backup path can be used to carry low priority traffic. The traffic affected after a failure occurs on the primary path is immediately switched at the source and the destination nodes to the backup path.

Obviously, the restoration time is faster in the $1+1$ configuration than in the $1:1$ configuration because in the $1+1$ configuration, there is no need to synchronize between the two switches at end-nodes of the path as in $1:1$ configuration. However, $1:1$ configuration allows the low priority traffic to be carried on backup paths in normal operation.

Shared protection

Shared link protection allows different backup paths to share one or more wavelength channels if the corresponding primary channels are allocated on different links. Similarly, shared path protection allows different backup paths to share one or more wavelength channels if the corresponding primary paths are disjoint (link-disjoint or node-disjoint).



(a) *Shared link protection*

(b) *Shared path protection*

Figure 2.7: *Shared protection schemes*

Figure 2.7 shows examples of the two schemes of protection. Figure 2.7(a) shows a scheme of shared link protection. The two backup paths $r_1 = (5 \rightarrow 2 \rightarrow 3)$ and $r_2 = (2 \rightarrow 3 \rightarrow 6)$, which protect a primary channel on link $(5-3)$ and a primary channel on link $(2-6)$, share wavelength λ_1 on link $(2-3)$. Note that these two primary channels are on different links. The example in Figure 2.7(b) shows a shared protection scheme. The two backup paths $r_1 = (1 \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 3)$ and $r_2 = (5 \rightarrow 2 \rightarrow 6 \rightarrow 4)$ of the corresponding primary paths $p_1 = (1 \rightarrow 2 \rightarrow 6)$ and $p_2 = (5 \rightarrow 3 \rightarrow 4)$ share λ_2 on link $(2-5)$. We note that p_1 and p_2 in this example are node-disjoint.

In shared protection schemes, the efficiency of capacity utilization depends on a *shared factor*. The shared factor is defined as the number of backup channels, Q, which share one allocated channel for $1:Q$ configuration or share P wavelength channels for

$P : Q$ configuration.

Table 2.2: *Dedicated protection versus shared protection*

Dedicated Protection	Shared Protection
• Fast restoration • Capability of supporting multiple failures	• Better resource utilization • Limited support for multiple failures

Dedicated and shared protections are two common schemes that can serve different purposes of protection. For example, as in Table 2.2, dedicated protection scheme is usually preferred when protection requirements fast restoration against multiple failures and 100% restorability. On the other hand, although shared protection can also assure 100% restorability against single link failures, it is preferred in protection schemes where capacity utilization is a major consideration, but not for high priority of protection.

2.4 Existing approaches to the SLTD problem

Path/link protection and dedicated/shared protection are key schemes for network survivability at logical topology design level. The implementation for these schemes is often based on ILP formulation [10, 12, 9, 11] and graph theory [13, 14, 15, 16]. In this section, we discuss these two approaches and briefly discuss the strengths and weaknesses of these approaches.

2.4.1 ILP formulation approach

The SLTD can be modeled through an exact formulation, known as ILP formulation. This mathematical model, in general, is used to obtain the exact (optimum) solution to optimization problems. Hence, ILP formulation attempt to find the optimum (minimum or maximum) value of an objective function subject to some constraints.

For the SLTD problem, research community has usually investigated network survivability through two popular objectives. One objective is to minimize the total capacity (wavelength channels) usage in the network. This objective is employed for capacity utilization purposes. Another objective that is used for load balancing purposes is to minimize the maximum capacity used on fiber links, referred to as *network congestion level*. These objectives are subject to different constraints such as the flow conservation constraint, the capacity constraint, survivability constraint, and other mathematical constraints (eg: integer constraint, range of variables, etc). These constraints are described below.

- *The traffic flow conservation constraint*

  A traffic requirement from source node s to destination node d is routed through paths from s to d. Such paths is referred to as a flow from s to d. The direction of the flow is unknown before it is established, and hence a constraint, namely the flow conservation constraint, is necessary to ensure the integrity of the flow. In WDM networks, a flow is a lightpath between a source node and a destination node, that is a sequence of nodes and links from s to d. The flow conservation constraint refers to the relationship between the total incoming and outgoing flow in each node of the network. The relationship differs amongst three types of nodes in the network: source node, intermediate nodes and destination node, and is stated as follows:

  - The total flow out of a source node must be larger than the total flow into the source node by the number of capacity units required for this connection.
  - The total flow out of a destination node must be less than the total flow into the destination node by the number of capacity units required for this connection.
  - The total flow out of an intermediate node must be equal to the total flow into the node.

- *The capacity constraint*

  The capacity constraint requires the total flow passing through a link to be less than or equal to the total capacity available in that link.

- *Survivability constraint*

  This constraint assures that backup paths and the corresponding primary paths are disjoint (link-disjoint or node-disjoint).

- *Integer constraint*

  In ILP formulation, a network flow is modeled through decision variables. For example, in order to identify if a flow f traverses though a link e_i, a decision binary variable w_i^f is employed and implies a portion of the flow f traverses through link e_i if w_i^f is larger than 0. Since a flow in SLTD is defined as a lightpath, the decision variables in ILP formulation have to be integer numbers (usually binary numbers).

ILP formulation is considered as a standard solution for protection schemes [11, 12, 17, 10]. This is because ILP formulation can yield optimum solutions and, more importantly, different protection schemes can be joined and modeled in an ILP formulation. These models are developed according to the different context of the network. For instance, in [11], protection schemes (path/link protection and shared/dedicatedprotection) are modeled for simple networks in which a link contains only one optical fiber. On the other hand, Hui Zang *et al.* [12] investigated path protection in a more complicated physical mesh architecture called "Duct layer" or "Shared Risk Link Group". In this network architecture, path protection require the primary paths and the corresponding backup paths not to be link-disjoint, but duct-disjoint so that the network is survivable under single-duct failures. Similar results have been achieved through different studies and summarized as follows:

- Path protection offers much better capacity saving over link protection [11, 10]. On the other hand, path protection is more susceptible to multiple link failures than link protection.
- A similar result is achieved between shared and dedicated protection schemes, that is, shared protection offers better capacity utilization over dedicated protection. In contrast, shared protection is more susceptible to multiple link failures than dedicated protection [11].
- These results are achieved through a unified ILP formulation of protection in [10]. Authors found that more than 160% of additional capacity is required for dedicated path protection, while, for shared path protection, the total capacity of primary and protection traffic is only 165%. The required additional capacity for dedicated link protection is over 329% but reduces to 74% for shared link protection.

Joint optimization in which problems of routing and wavelength assignment is simultaneously modeled and solved is intractable with even very small networks since the number of decision variables and constraints is increase quickly. Some heuristic approaches have been proposed to reduce the number of decision variables and constraints [18] or to relax the integer constraint [19].

In [18], the joint optimization is decomposed into two subproblems. The decomposition is an approximation in the sense that solving the subproblems in sequence and combining the solutions may not result in the optimal solutions for the jointed optimization problem, or the final subproblem may have no solution from the earlier subproblem's results even if the original jointed problem contains a solution. The subproblems are as follows:

- **Lightpath routing subproblem:** This subproblem aims to determine the primary paths and backup paths for connections.
- **Wavelength assignment subproblem:** The aim of this subproblem is to assign wavelengths to primary paths and backup paths resulting from lightpath routing subproblem.

The number of decision variables and constraints for ILP formulation in the each subproblem is obviously much less than those in the original formulation. For example, in the network of W wavelength per links, the number of decision variables in lightpath routing subproblem is W times less than the original problem. However, the lightpath routing or the wavelength assignment subproblems itself is also an NP-complete problem [5, 20, 21], ie. they are still intractable with large scale networks. Another approximate solution for ILP is based on *randomize rounding* was proposed in [19]. This approach includes three steps: non-integral multicommodity flow, path stripping and randomization. Non-integral multicommodity flow relax $0-1$ integer formulations and solves the relaxed problem by linear programming. Path stripping converts the edge flows of commodity i to a set of path τ_i that may carry the flow commodity i in the optimal case. Finally, randomization selects a suitable path for commodity i from τ_i by casting a $|\tau_i|$ dice with face probability equal to the weights of the paths in τ_i. This approach provides a fast engine for ILP solution but the solution is near-optimum and non-deterministic.

2.4.2 Graph theory approach

ILP formulation is preferred for obtaining optimum solutions but it is intractable with large scale networks. Graph theory approach based on graph algorithms is a possible alternative solution for the SLTD problem. For optimization reasons, shortest path algorithms and k-shortest paths algorithms are usually employed to establish lightpaths. In addition, under network survivability context, two-step approach and one-step approach are two common graph algorithms employed to find two disjoint paths between any two nodes in the network. These approaches are based on shortest path algorithms and stated as follows:

1. *Two-step approach*

   Two step approach aims to find disjoint path-pairs in which primary paths and backup paths are discovered separately. In the first step, the primary path is determined over the original physical network. Then, all links contained in the primary path are removed out of the network. In the second step, the backup

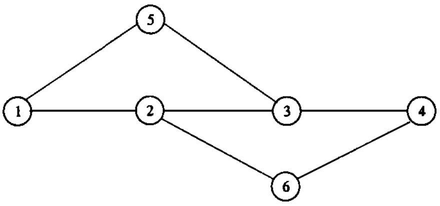


Figure 2.8: *The trap topology*

path is determined over the residual network. Both primary path and backup path are usually determined using shortest path algorithms such as Dijkastra's algorithm or Bellman-Ford [22, 23].

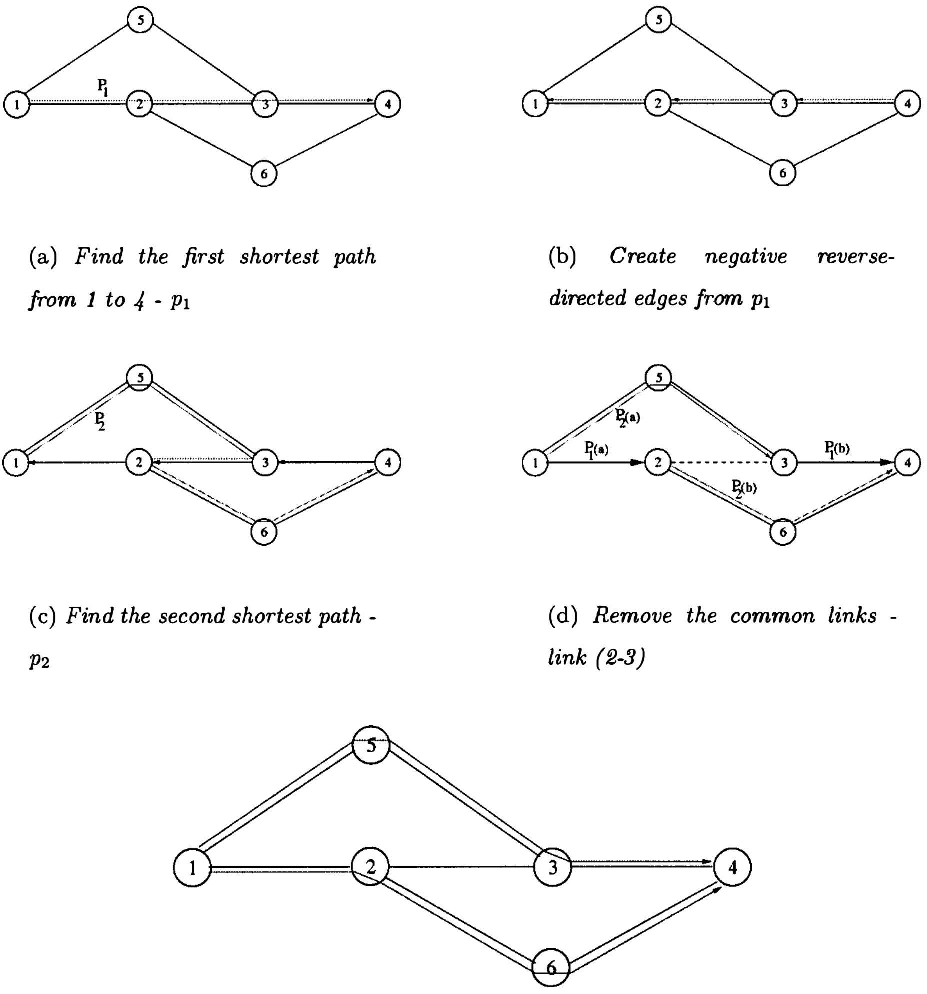
2. *One step approach*

   Two step approach is simple in both idea and implementation. However, there is no guarantee that the total cost of the found pair of disjoint paths is minimum. In addition, this approach may fail in some network topologies such as the "trap topology" shown in Figure 2.8. For example, the first path found between source node 1 and destination node 4 in Figure 2.8 may be ($1 \dashrightarrow 2 \dashrightarrow 3 \dashrightarrow 4$), remarked as the primary path of connection $(1,4)$. The backup path of the connection is determined after all links contained in the primary paths (links $\{(1-2),(2-3),(3-4)\}$) are removed. It shown can easily be seen that the backup path can not be found since the network is disconnected between node 1 and 4.

   One step approach - as the name suggests - implies determination of a primary path and a backup path simultaneously. The algorithm for this approach is proposed by Surballe, namely *Surballe's algorithm* [24]. Bhandari [25] modified Surballe's algorithm to adapt with negative weight of links. This algorithms resolves the above disadvantages of two-step approach and is stated in [3] as follows:

The two-step and one-step approaches are simple and computationally efficient. However, since these approaches provision traffic connections sequentially without backtracking, sometimes no solution may be found even when a feasible solution does exist. In fact, the graph theory approach is more applicable in online provisioning than in offline provisioning. These approaches have been extensively studied in the literature [15, 16, 26, 27].

In [15], authors investigated survivable routing based on two-step and one-step approaches, namely *Separate Path Selection* (SPS) and *Joint Path Selection* (JPS) respectively. These approaches aim to optimize the network resource utilization of each connection by minimizing the total cost of the primary and backup paths. The



(a) *Find the first shortest path from 1 to 4 -* p_1

(b) *Create negative reverse-directed edges from* p_1

(c) *Find the second shortest path -* p_2

(d) *Remove the common links - link (2-3)*

(e) *A pair of link-disjoint paths is created*

Figure 2.9: *One step approach - Bhandari's algorithm*

Algorithm 1 *One-step approach*

```
1: Take a shortest path between the source node $s$ and the destination nodes $d$. Denote this as $p_1$ (Figure 2.9(a)).
2: Define the direction of each edge traversed in $p_1$ from $s$ toward $d$ as positive.
3: Remove all directed edges on the shortest path $p_1$ and replace them with reverse direction edges by multiplying -1 to the original edge cost (Figure 2.9(b)).
4: Find the least cost path from $s$ to $d$ in the modified graph using the modified Dijkstra's algorithm. Denote this path as $p_2$(Figure 2.9(c)).
5: Remove any edge of the original graph traversed by both $p_1$ and $p_2$ (Figure 2.9(d)). These are called *interlacing edges*. Identify all path segments remaining after process of the edge removal (Figure 2.9(e)).
```

performance of the approaches is evaluated for different protection path cost functions. The results have shown that JPS is substantially better than SPS and also scales very well in terms of the network resource abundance. In addition, JPS can utilize network resources better than SPS. In [16], two-step and one-step approaches are also investigated, but with a different objective function based on the blocking probability. The simulation results have shown that the one-step approach significantly outperforms the two-step approach. The improvement in blocking probability is significant, around 10%-20%, especially when fixed routing is used.

2.5 Concluding Remarks

The performance of the SLTD problem depends on two key factors: the survivability scheme and the implementation of the scheme.

Different survivability schemes use a variety of performance measures such as resources utilization and restorability. Path protection and shared protection schemes are preferred for better network resource utilization whereas link protection and dedicated protection schemes are used for fast restoration. Each protection scheme has its own advantages and disadvantages and is applicable to circumstances, hence many protection schemes have been proposed to adapt to the desired performance objectives. The *path segment protection* [28, 29] and *p-cycles* [30, 3, 31, 32] are to most significant schemes to enhance the performance of path and link protection. These schemes, however, are not the focus of this thesis.

The performance of each protection scheme can be implemented through two common approaches, namely ILP formulation and graph theory approach. ILP formulation offers a useful mathematical tool to obtain optimal solutions. However, this approach is intractable with even for moderate scale networks. In fact, the SLTD problem has been proven to be NP-hard. In contrast, approaches in graph theory such as the two-step and the one-step approaches, are computationally efficient, but they may not find a solution in some network topologies such as "trap topology" and result in a very high congestion level. In addition, no feasible solution may be found even that a solution may exist.

In this thesis, we attempt to solve the SLTD problem in moderate scale networks by combining the computational advantages of the graph algorithms and optimal solutions of the ILP solver with small integer variables and constraints.

Chapter 3

Survivability of Physical Topology

As we have mentioned in Chapter 2, existing approaches to network survivability have usually overlooked the physical topology of the network, or have assumed that the physical topology is guaranteed to be survivable. This assumption is very strong and usually causes a so-called redundant-checking dilemma. That is if physical topology is not survivable, then seeking for survivability at logical topology design is clearly redundant. It is, therefore, imperative and crucial to address the survivability at the physical topology as the first step before considering logical topology design.

In practice, network traffic requirements are different from application to application. For example, a broadcasting application may need no protection for the data, while communication applications in military may require a very high level of protection because of the importance of the transmitted data. The concept of survivability at physical topology, therefore, is different dependent on the levels of protection required in logical topology design. A physical topology is called survivable if it is two-connected. In other words, there is at least two disjoint paths, termed as *disjoint path-pair*, between any two nodes in the network. A path-pair is called a *link-disjoint path-pair* if its two paths do not share any edge. Likewise, it is called *node-disjoint path-pair* if the two paths in the pair do not use any node in common.

A popular assumption, found in the literature, to ensure the survivability at the physical network has been based on the size of cut-sets, which we shall refer to as the *cut-set assumption.* The cut-set assumption, however, does not necessarily imply a network topology is two-connected. In fact, a high protection requirement at logical topology design may fail in some cases of physical networks that even satisfy this assumption. In addition, although implementation of this assumption is not complex, the computational time is expensive with large scale networks because of the explosion of cut-sets. It is, therefore, necessary to have a further identification of the physical topology with fast computation to compromise protection requirements at logical topology design. Except explicitly stated otherwise, throughout this chapter, the term *network survivability* implies the survivability at the physical network.

In this chapter, we provide an in-depth analysis of existing methods in the literature towards the problem of network survivability. Our aim is to dissect this body of work, pointing out the weaknesses and open problems in this field. Next, we propose a novel approach to determine if a network topology is unsurvivable, link-survivable, or node-survivable. Our contributions from this chapter are twofold. First, we provide a unified view on physical topologies with respect to the problem of network survivability. Secondly, a new method is proposed for this problem in order to resolve the above commitments of cut-set assumption for network survivability.

The rest of this chapter is organised as follows. We introduce and define terminology for physical topology with respect to survivability view in Section 3.1. We review and analyse the body of existing works in Section 3.2. Finally, our approach to determine the exact survivability configuration of physical topology with polynomial time is proposed in Section 3.3

3.1 Problem Setting

As we have identified, protection requirements at logical topology design differ from application to application. In [3], Wayne D. Grover classifies protection requirements according to Quality of Protection (QoS). Traffic connections are routed in one of the following protection classes: economy (preemptible services), bronze (no-protection services), silver (best-efforts restoration), and gold (assured restoration). Solving protection requirements with respect to this classification, however, is purely at logical topology design without any relation to physical topology. The protection requirements in our study, in different way, is classified according to network component failures, ie: link failures and node failures. The influence of link failures to network operation is not as serious as node failures because a failure of a network node is equivalent with failures of all links connected to this node. Therefore, based on different levels of protection required at the logical layer, we classify physical topology into four classes: *node-survivable topology*, *link-survivable topology*, *unsurvivable topology*, and *unconnected topology*.

- *Node-survivable topology* can offer at least one node-disjoint path-pair between any two nodes in the topology. Since a node-disjoint path-pair is also link-disjoint path-pair, the topology can be designed to respond the failures of variety of network elements such as network nodes and network links. Thus node-survivable topology provides the highest level of security.

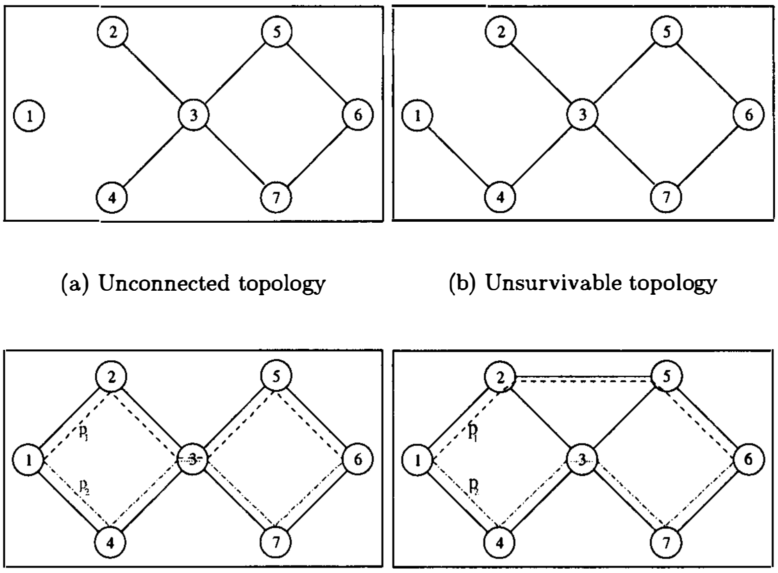
- *Link-survivable topology* is able to offer at least one pair of link-disjoint-paths between any node-pair in the network. This class of physical topology can be employed to support protection mechanisms against link failures such as fiber cuts.

- *Unsurvivable topology* is a connected topology that does not satisfy the conditions of node-survivable and link-survivable. In such a topology, a failure of a link may cause the network to be disconnected. Thus, this configuration only offers low-priority protection or no protection at all.

- *Unconnected topology* contains at least one node of zero degree. Such topology does not exist in real network systems because no service provider requires a network node designed to do nothing.

Figure 3.1 illustrates the four classes of network topology. Topologies are constructed from unconnected to node-survivable. An unconnected topology is shown in Figure 3.1(a). Since degree of node 1 in the topology is equal to 0, it is classified in unconnected class; the unsurvivable topology in Figure 3.1(b) is created by connecting node 1 with node 4. In Figure 3.1(c), the link-survivable topology is completed by linking node 1 with node 2. The pair of two paths (p_1, p_2) in Figure 3.1(c) is link-disjoint because they do not share any physical link. In addition, it can be seen that p_1 and p_2 share node 4, therefore the path-pair is not node-disjoint. In other words, the physical topology in Figure 3.1(c) is link-survivable. Link-survivable topology are provided to deal with links failures such as fiber cuts. Backup paths in this topology are required to be link-disjoint with their corresponding primary paths, but they are not necessarily node-disjoint. The topology in Figure 3.1(d) is a modified version of topology in Figure 3.1(c) by adding link $(2, 6)$ to the graph. Evidently, this topology can offer pairs of node-disjoint paths between any node-pair in the network. Such topology is referred to as node-survivable networks.

In short, a node-survivable network can offer both node-disjoint paths and link-disjoint paths, but a link-survivable network can only offer link-disjoint paths between all node-pairs and can offer some node-disjoint path-pairs but not all. Table 3.1 presents the classes of physical topology to support logical topology design with different types of



(a) Unconnected topology

(b) Unsurvivable topology

(c) Link-survivable topology

(d) Node-survivable topology

Figure 3.1: *Physical topology classification*

survivability requirements. It can easily be seen that node-survivable topology support all requirements of logical routing design with protection of node failures, link failures and non-protection; link-survivable topology is for link failures and non-protection requirements; and unsurvivable topology can only support the logical routing with non-protection.

Table 3.1: *Physical topology classes and their support for different survivability requirements*

		TYPES OF PHYSICAL TOPOLOGY		
		Unsurvivable	Link-survivable	Node-survivable
LOGICAL TOPOLOGY	Non-protection	✓	✓	✓
	Link failures	×	✓	✓
	Node failures	×	×	✓

In coming sections, the terminology taken from graph theory is regularly used. Hence, for clarity of discussion, we introduce the terminology here.

- A physical topology is represented by a ***graph*** and denoted as $G(V, E)$, where V is the set of network nodes and E is the set of network links.
- A graph is called ***connected*** if any two of its nodes are linked by a path.

- A maximal connected sub-graph (or sub-topology) is called a **component**.
- A physical topology is **survivable** if it is link-survivable or node-survivable.
- A node is called ***node-bridge*** or ***articulation node*** if it is a unique common node of two node-survivable components.
- ***Link-bridge*** is a link that connects two survivable subnetworks.

3.2 Existing Approaches to Survivable Networks

The survivability of a network can be verified manually or automatically. Manual verification is very suitable with small networks where designers can perform the verification just in few seconds or few minutes. However, manual verification may take hours or days for large scale networks. Furthermore, it is prone to human errors, particular with tasks that requires a long period of working. Therefore, automatic survivability verification is important in both theory and practice. The concept of survivable networks is more complex than the concept of connectivity in graph theory. In addition, efficient automation algorithms based on graph theory can help designers to save the computational time and avoid human errors. In this section, we investigate the strengths and weaknesses of a popular assumption used in research community, namely *cut-set assumption.* We, however, first distinguish the difference between 2-connected definition and node-degree of two.

3.2.1 Two-connected versus node-degree of two

A network in which the degree of every node is equal or larger than 2 is denoted to be node-degree of two. *A network is survivable if it is two-connected.* At a glance, this definition sensitively leads to a view that the network is node-degree of two. Since every node is connected with at least two other nodes in the network, they seem to be able to offer two disjoint paths between any two nodes in the network. In fact, this is a misconception in two-connected concept, or survivability characteristic of networks. If a network is survivable (two-connected) then node degree of all nodes in the network is equal or larger than 2; but a network in which the degree of all its nodes equal or larger than 2 is not always survivable. The topology in Figure 3.2 illustrates this misunderstanding. Path $(3-5-6)$ is a bridge that connects two subsets of network nodes $X = \{1,2,3,4\}$ and $Y = \{6,7,8,9\}$. As a result, all paths between nodes $x \in X$ and $y \in Y$ must share the same path $(3-5-6)$. Hence, even network has all node with degree equal or larger than 2, it is not a survivable network.

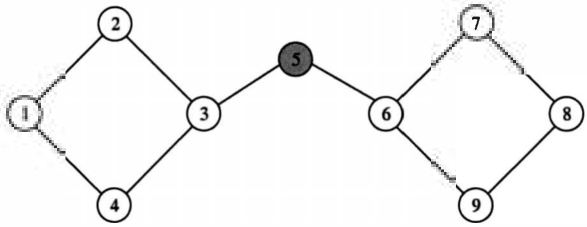


Figure 3.2: *An illustration of nodal degree failure*

Therefore, all algorithms for verification network survivability based on node-degree of two may yield undesirable results and hence they are not reliable. Cut-set assumption below have been preferred for the accuracy of network survivability verification.

3.2.2 Cut-set assumption

A network is survivable if the size of every cut-set of the network is equal or larger than 2.

Let $G = (V, E)$ be a network topology. A cut in G is a partition of V into parts S and $\bar{S} = V - S$. Each cut defines a set of edges consisting of those edges in E with one end-point in S and the other in $\bar{S}$. This edge set is referred as the cut-set $CS(S, V - S)$ associated with the cut $\langle S, V - S\rangle$. Let $|CS(S, V - S)|$ be the size of the cut-set, $|CS(S, V - S)|$ is the number of links between S and $V - S$. Thus, according to the cut-set assumption, *a network is survivable if* $|CS(S, V - S)| \geq 2, \forall S \subset V$. If S is a subset of only a single node in the network, then cut-set assumption is essentially the same as the node-degree assumption.

Since cut-set assumption is in touch with the number of links connected between two subsets of a cut, the assumption can assure the network to offer *link-survivable*, but *not node-survivable*. In other words, a configuration of the network that satisfies the condition of cut-set assumption can provide at least one link-disjoint path-pair between any distinct pair of source node and destination node.

The implementation of the assumption is not complex but the computational time with large scale networks is its largest disadvantage. The number of cut-sets is exponentially increase with the increasing of network nodes and is calculated as below [3].

$$N_{cutset} = 2^N - 2$$

where N_{cutset} is the number of cut set in the network, N is the number of network nodes.

Table 3.2 shows the example of number of the possible cut-sets versus N_{cutset} the number of network nodes N. The number of cut-set is double with the unit increasing of network nodes. For instance, N_{cutset} in a network of 20 nodes is over 1 million; the figure is over 32 million cut-sets with $N = 25$ node networks, $32(= 2^5)$ times larger than $N = 20$; and the number of cut-sets in the networks of $N = 30$ nodes is up to 1 billion cut-sets. So, cut-set assumption becomes intractable even with moderate scale networks ($20 \leq N \leq 30$).

Table 3.2: *The number of cut-sets versus the number of network nodes*

N	20	25	30
N_{cutset}	$1,048,574$	$33,554,430$	$1,073,741,822$

In summary, the node-degree assumption is simple but not reliable for verification of survivable networks. Meanwhile, the cut-set assumption is only applicable for link-survivable networks, and it is intractable with large scale networks. The verification ability of two these assumptions into different classes of physical topology are summarized in Table 3.3. The node-degree assumption can not determine any type of physical topology whereas the cut-set assumption can verify whether a network is survivable or not, but the cut-set assumption can not identify exactly a link-survivable topology or a node-survivable topology. In the next part, we propose an approach that can classify network topologies, and determine if they are unconnected, unsurvivable, link-survivable or node-survivable.

Table 3.3: *Performance of two common assumptions over different classes of physical topology*

	PHYSICAL TOPOLOGY	
	Unsurvivable	Survivable
Node-degree assumption	×	×
Cut-set assumption	✓	✓

3.3 Survivable-based Approach for Verification of a Survivable Network: Theoretical Analysis

In general, an arbitrary network topology comprises of distinct subnetworks that may be node-survivable, link-survivable, unsurvivable or unconnected. In this section, we first propose a theory to understand an arbitrary topology. Next we implement the

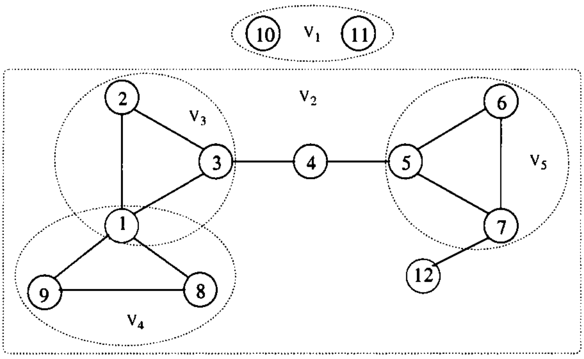


Figure 3.3: *An arbitrary topology*

theory into the algorithm, called as ***survivable-bases***, that automatically verifies and results in the specific class of subnetworks. This approach avoids human errors in manual verification and the time complexity of cut-set assumption, and is capable of verifying different classes of topology.

For clarity, we define some survivable graph terminology as follows. *Unconnected nodes* are the nodes of zero degree; *unconnected subnetwork* is set of unconnected nodes; *unsurvivable subnetworks* are the maximum connected subnetworks; *link-survivable subnetworks* are the subnetworks in which there exists at least one pair of link-disjoint paths between any two nodes of the subnetworks; and *node-survivable subnetworks* are the subnetworks in which there exists at least one pair of node-disjoint paths between any two nodes of the subnetworks. For example, In Figure 3.3, nodes $\{10, 11\}$ are unconnected nodes and set V_1 of the nodes is an unconnected subnetwork; V_2 is a unsurvivable subnetwork; V_3, V_4 and V_5 are node-survivable subnetworks; and $V_3 \cup V_4$ is a link-survivable subnetwork.

3.3.1 Node-survivable networks

Given a physical topology $G(V, E)$, a subnetwork or subtopology is a subgraph of G and denoted as $G_s(V_s, E_s)$, where V_s is a subset of V and E_s is a subset of edges E in G_s. Since the survivability properties of networks and subnetworks are the same, all our discussion in networks is applicable for subnetworks as well.

By definition, $G(V, E)$ is *node-survivable* if and only if there exists at least one node-disjoint path-pair between any two nodes of the network. Thus, a subnetwork $G_s(V_s, E_s)$ is node-survivable if and only if there exists at least one node-disjoint path-pair between any two nodes of the subnetwork. This definition will be used as the principle to prove our theory. For convenience, the term "physical topology" is used to imply either "network topology" or "subnetwork topology". The following theorem proposed by us forms the foundation of our methods to establish survivability of networks.

Theorem 3.3.1. *A physical topology is node-survivable if it contains an Euler cycle.*

Proof. Assume $G(V, E)$ is the physical topology, then $G(V, E)$ contains an Euler cycle C_E. Since C_E is a cycle that contains all nodes in G, two node-disjoint paths can be found between any two nodes (s, d) in the topology. One path is from s and travels forward to clockwise direction of C_E to destination node d and another path is forward to anti-clockwise direction of C_E. This topology, therefore, is node-survivable. □

Theorem 3.3.1 assures that if a topology contains an Euler cycle, then it is node-survivable. However, some node-survivable topologies, such as that in Figure 3.4, may not contain an Euler cycle.

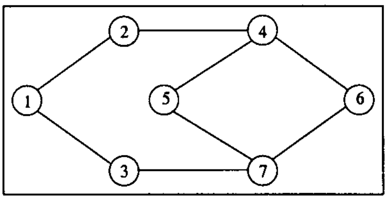


Figure 3.4: *Node-survivable topologies: Non-Eulerian topology*

Theorem 3.3.2. *A physical topology is node-survivable if and only if it can be constructed from a cycle by successively adding* ***survivable-paths*** *to node-survivable topologies already constructed.*

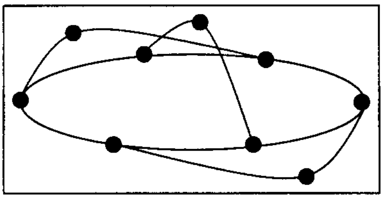


Figure 3.5: *Node-survivable topologies: survivable paths' construction*

Proof. A ***survivable-path*** is a non-trivial path that has its end-nodes belonging to node-survivable topologies. Clearly, every topology constructed as in Theorem 3.3.2 is node-survivable. Conversely, let G be a node-survivable topology. Then G contains a cycle, and hence has a maximal subgraph (or sub-topology) H constructible as above. Since any edge $xy \in E(G)\backslash E(H)$ with $x, y \in H$ would define a survivable-path, H is an induced subgraph of G. Thus if $H \neq G$, then by the connectedness of G there is an edge vw with $v \in G - H$ and $w \in H$. As G is a node-survivable topology, $G - w$ contains a $v - H$ path P. Then wvP is a survivable-path in G, and $H \cup wvP$ is constructible subgraph of G larger than H. This contradicts the maximality of H. □

The theorem states the necessary and sufficient condition to verify or construct a node-survivable topology. If a physical topology satisfies the condition of the theorem, then it is node-survivable. Otherwise, the topology should be in another class of physical topology such as link-survivable, unsurvivable or unconnected.

3.3.2 Link-survivable networks

A link-survivable network is a networks in which there exists at least one link-disjoint pair-path between any node-pair in the network.

Obviously, a node-survivable topology is also a link-survivable topology because a node-disjoint path-pair is also a link-disjoint path-pair. Conversely, a link-survivable network is not always a node-survivable topology. However, a link-survivable network can be constructed from node-survivable subnetworks.

Lemma 3.3.3. *A link-survivable topology can be constructed from two node-survivable topologies that contain exactly one node in common.*

Proof. Assume G is a graph that is formed from two node-survivable subnetworks G_1 and G_2 and $x \in G$ is the unique common node of G_1 and G_2. We prove that this graph is link-survivable. First, since G_1 is a node-survivable topology, there exists at least one pair of node-disjoint paths, which is also a pair of link-disjoint paths, between any two nodes in the topology. It also has a similar explanation for node-pairs in topology G_2. We prove that there always exists at least a pair of link-disjoint paths between every node-pair $v \in G_1$ and $w \in G_2$. Since node-pair $(v, x) \in G_1$, there exists at least one pair of node-disjoint paths (p_{11}, p_{12}) between this node-pair. Similarly, there exists at least one pair of node-disjoint paths (p_{21}, p_{22}) between node-pair (x, w). Let p_{1vw} and p_{2vw} be the combined paths of $p_{11} - p_{21}$ and $p_{12} - p_{22}$, respectively. The path-pair is link-disjoint, and hence G is the link-survivable topology. □

Lemma 3.3.4. *A link-survivable topology can be constructed from a node-survivable topology and a link-survivable topology that contains exactly one node in common.*

Proof. Similar proof as Lemma 3.3.3. □

3.3.3 Unconnected and unsurvivable networks

Unconnected networks are networks that contain at least one node of zero degree.

On the other hand, unsurvivable networks are defined as connected graphs that do not satisfy the conditions of node-survivable and link-survivable. Obviously, node-survivable networks or link-survivable networks are connected graphs but are not classified as unsurvivable networks. The problem of verification of unsurvivable networks can be considered as the problem of connectivity in graph theory. However, we consider the problem with survivability perspective. A connected graph in general case can not support any survivable routing from logical topology requirements. In fact, this configuration is suitable with routing problems that do not require any protection. Such topology is considered *unsurvivable.*

Lemma 3.3.5. *Every unsurvivable graph contains a normal spanning tree, with any specific node as its root.*

Proof. Let G be an unsurvivable graph and $r \in G$ any specific node. Let T be a maximal normal tree with root r in G; we show that $V(T) = V(G)$.

Suppose not, and let C be a component of $G - T$. As T is normal, $N(C)$ is a chain in T. Let x be its greatest element, and let $y \in G$ be adjacent to x. Let T' be the tree obtained from T by joining y to s; the tree order of T' then extends that of T. We shall derive a contradiction by showing that T' is also normal in G.

Let P be a T'-path in G. If the ends of P both lie in T, then they are comparable in the tree-order of T (and hence in that of T'), because then P is also a T-path and T is normal in graph G by assumption. If not, then y is one end of P, so P lies in C except for its other end z, which lies in $N(C)$. Then $z \leq x$, by the choice of x. For our proof that y and z are comparable it thus suffices to show that $x < y$, i.e. that $x \in rT'y$. This, however, is clear since y is a leaf of T' with neighbor x. □

3.4 Proposed Survivable-based Algorithms for Physical Topology

Our approach to physical topology analysis uses the theory presented in the last section to investigate survivability of a given arbitrary topology. We aim to determine the class of a topology, identify subgraphs in the topology that are node-survivable, link-survivable, unsurvivable and unconnected. Our examined approach consists of two major steps (Figure 3.6). In the first step, the connectivity of a given topology is considered based on spanning tree algorithms. In the second step, survivability of the given topology is examined using a new algorithm called ***survivable-bases*** algorithm.

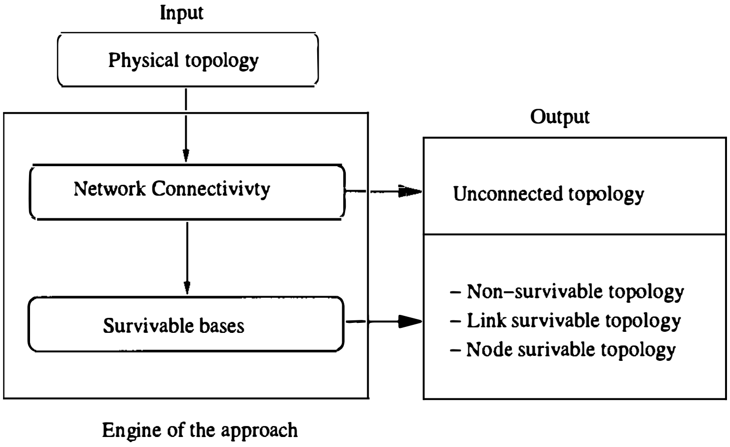


Figure 3.6: *Our proposed approach to physical topology classification*

3.4.1 The connectivity of physical networks

The connectivity of physical networks implies whether a topology is connected or not. In particular, this allows us to identify the class of unconnected topologies. In our approach, the connectivity of a topology is determined through a "forest" of maximal trees and follows two principles as below.

1. If a topology is constructed from one and only one maximal tree, then the topology is connected.

2. If a topology is constructed from more than one maximal tree or from one maximal tree and unconnected nodes, then the topology is unconnected.

The forest of maximal trees can be found by adopting prevalent minimum spanning tree algorithms in graph theory such as Prim's algorithm and Kruskal's algorithm [22].

These algorithms try to find in the graph the spanning tree whose total weight of edges is minimum. However, with respect to the connectivity of network topology, the weight of network links is not important and hence is not taken into account. In other words, we assume all network links to have the same weight. The Prim's algorithm and Kruskal's algorithm are modified to adapt with this assumption. In addition, with the modified algorithms, the search to find minimum cost is ignored and hence the computational time is improved.

Modified Prim's algorithm

Prim's algorithm is a greedy algorithm for obtaining a minimum-cost spanning tree; the next edge to include is chosen so that it results in a minimum increase in the total cost of the edges included so far. Let T be the set of edges selected so far; T forms a tree. The next edge xy to be added in T is a minimum cost edge not in T with the property that $T \cup xy$ is also a tree. Searching time for such edge xy take a significant part of computational time compared to the rest of the algorithm. Thus, in our algorithm, we employ the same idea but the searching process for minimum added edge can be relaxed to a simpler search to obtain tree T.

The pseudo-code of the modified Prim's algorithm is presented in Algorithm 2. This algorithm can be explained as follows. A tree T_1 is initialized with arbitrary link e in the graph G. Next, all neighbours of the end-nodes of edge e are marked and added to tree T_1. During the process of adding, the neighbours of added nodes are also marked for the next addition. The rule of marking is that if a node that is not in T_1 is a neighbour of more than one node in the tree, then this node is marked to be the neighbour of only one of these nodes. The process of edge adding is repeated until all nodes are added into the tree. In case that this graph is not connected, T_1 is the first tree found. The next tree $T_i, i = 2, 3, \ldots$ can be found by the successive application of the above process to subgraphs $G_i \leftarrow G_{i-1} - T_{i-1}$.

The complexity of the modified algorithm, the modified Kruskal's algorithm and others will be discussed in Section 3.4.3 where the computational time of our approach is evaluated.

Modified Kruskal's algorithm

The key idea of Kruskal's algorithm is based on a forest of trees; and if there exists edges in the graph that connect these trees into only one tree, the spanning tree is

Algorithm 2 *Modified prim's algorithm*

```
Input : Graph $G(V, E)$
Output: A Forest of Spanning Trees $T$
1: Initialize forest $T$ by a tree $T_i \leftarrow e \in E$;where $i = 1$;
2: Label all neighbours of the end-nodes in tree $T_i$;
3: Add links that connect the neighbours with tree $T_i$;
4: Repeat steps 2 and 3 until $T_1$ has no neighbour;
5: If $\cup_{k=1}^{i} T_k$ contains all nodes of graph $G$ then the algorithm is terminated; otherwise
   go to step 6;
6: $G$ is assigned to be $G - T_i$; $i \leftarrow i + 1$; go to step 2;
```

found. This algorithm is also a greedy algorithm. The minimum cost of the spanning tree can be obtained by choosing from the least cost to the greatest cost of links. Our modified Kruskal's algorithm use the same idea as this, except that the process of finding minimum cost links is ignored.

The algorithm is implemented as follows. A forest with one tree of one edge is initialized by picking up any edge in the graph. A new edge is added into a constructed tree in the forest if it has one end-node in this tree and other end-node not in the forest. Two trees in the forest can be joined by a new edge if the edge has one end-node one tree and another end-node in the other tree. This reduces the number of trees in the forest. A new tree is created if both end-nodes of the new edge is not in the forest. The procedure is repeated until a spanning tree is obtained or there are no more edges in the graph to process. The algorithm produces a number of possible maximal trees in the graph. If this graph is connected then there will be only one spanning tree. The pseudo code of the algorithm is presented in Algorithm 3.

3.4.2 Survivable-bases algorithm

All unconnected physical topologies can be identified in the step described in Section 3.4.1. The input topology in the survivable-base step will be complete connected topologies or maximal connected subgraphs of the unconnected topology. In other words, the input physical topology in this step is assumed to be connected. The idea of this step is based on node-survivable subnetworks, called ***survivable-base***. According to the Theorem 3.3.2, a node-survivable subnetworks can be constructed from a cycle by successively adding **survivable-paths** to node-survivable topologies already constructed.

Algorithm 3 *Modified Kruskal's algorithm*

```
Input : Graph $G(V, E)$
Output: Spanning tree $T$
  Initialize forest $T$ by a tree $t_1 \leftarrow e_i \in E$;
  $E \leftarrow E - \{e_i\}$; $m \leftarrow 1$;
  while $(m < |V|)$ and $(E \neq \emptyset)$ do
    Pick a link $e \in E$;
    Check $e \in T$;
    if $e \notin T$ then
      Create a new tree $t$; $T \leftarrow T \cup t$;
      $m \leftarrow m + 1$;
    else if $e$ has one end-node in tree $t_k$ of forest $T$; then
      $t_k \leftarrow t_k \cup \{e\}$; $m \leftarrow m + 1$;
    else
      if $e$ has its end-nodes in two distinct tree $t_x$ and $t_y$ then
        $t_x \leftarrow t_x \cup t_y$;
        $T \leftarrow T - t_y$; $m \leftarrow m + 1$;
      end if
    end if
    $E \leftarrow E - \{e\}$;
  end while
```

Let $Sb = \{v_1, v_2, \ldots, v_{k+1}, \ldots, v_{k+n}, \ldots, v_d\}, d \geq (k+n)$ be a constructed survivable-base in the network. Survivable-paths can then be determined through cycles that contain them. Let $C = \{v'_1, v'_2, \ldots, v'_{k+1}, \ldots, v'_{k+n}, \ldots, v'_d\}$ be a cycle in the graph with $v'_i \equiv v_i,\ i \in \{(k+1) \ldots (k+n)\},\ n \geq 2$. Then, a survivable-path is contained in C and determined as

$$SP = \{v'_{k+1}, v'_k, \ldots, v'_1, v'_d, v'_{d-1}, \ldots, v'_{k+n+1}\}$$

The survivable-base is extended by adding such survivable-paths. This is also the key step in our survivable-base algorithm. Since a cycle itself is a survivable-base, our algorithm is operated over cycles. These cycles can be determined through the spanning tree and the set of remained links in the topology (Algorithm 5). ***The remained links*** are the links in the graph but not in the spanning tree.

The result of this algorithm is a new graph S that can be constructed through the following steps. The physical topology consideration is finished by the following principles:

Algorithm 4 *Survivable Graph S*

```
1: The first survivable-base is found by picking up a link in remained link set and joining it with a unique path from the corresponding source and destination. This survivable-base is modeled to be the first node $s_1$ in graph $S$.
2: The next survivable-base is combined into a constructed node $s_i$ if they contain more than one node in common. Otherwise a new node is added to graph $S$. If the new added node contain one node in common with any constructed node $s_i$, then they are linked by an edge. The weight of the edge is labeled by the value of common node.
3: Repeat step 2 until a node-survivable topology is resulted in or there are no more links to process.
4: Nodes in the constructed graph $S$ that form a cycle the weight of whose edges are not equal, are combined into a node. This is processed to all cycles in graph $S$. As a result, $S$ is only in the form of an unconnected graph or a tree.
```

- If graph S contains only one node, then the topology is node-survivable.
- If graph S is connected, then the topology is link-survivable and labels of links denote the node-bridges.
- If graph S is unconnected, then the topology is unsurvivable.

In the algorithm, we employ a procedure to find the cycles as survivable-bases. The detailed discussion below will explain how this procedure works and why the cycles have to be obtained from a spanning tree.

Finding cycles procedure

As discussed above, a cycle is found over a tree and an edge whose end-nodes are in the tree. All the edges are in the set of remaining links. Since there exists a unique path between any node-pair in a tree, the cycle so formed is unique. On the other hand, since the number of edges in a tree of N nodes is $N-1$, the searching method over a tree to find a path is not expensive in terms of complexity. Our approach employs the Breadth-First-Search (BFS) to find such a path over the tree. The steps are summarized in Algorithm 5. Note that if a cycle P is found, then P is represented as an "open cycle". The cycle is of the form $P = [s = v_1, v_2, \ldots, v_k = d]$. In such a cycle, the starting node and the ending node are not the same, but are connected together.

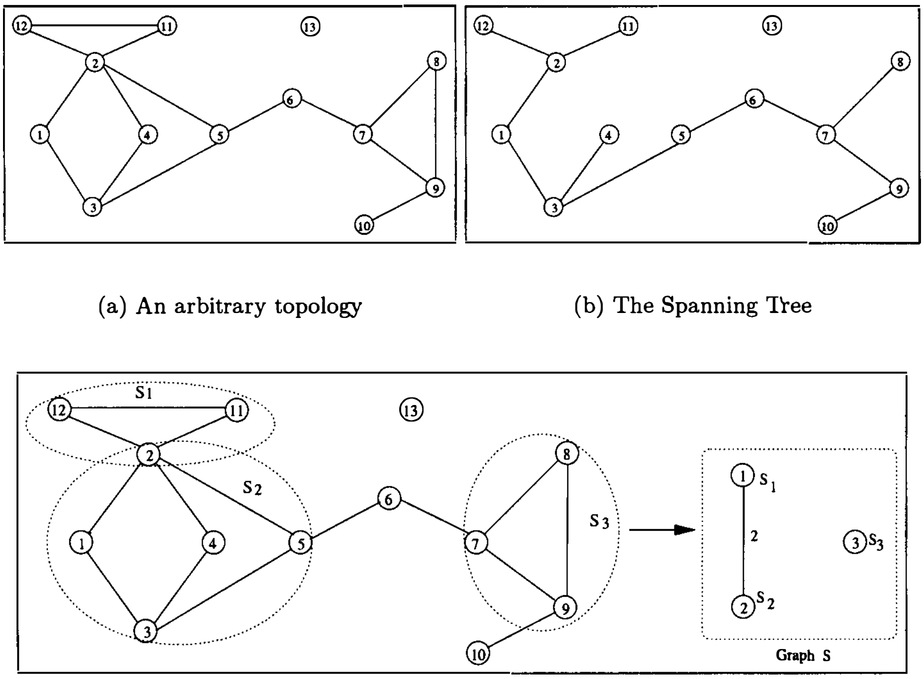
Algorithm 5 *Finding cycle*

```
Input : A tree $T$ and a edge $e$ whose end-nodes is in $T$;
Output: A cycle $P$ formed by $T$ and $e$;
  $(s, d) \leftarrow$ end-nodes of $e$;
  $queue \leftarrow [node.s, node.P]; check \leftarrow 0$;
  while $check == 0 \& queue \neq \emptyset$ do
    $[v] \leftarrow head(queue)$; $queue \leftarrow queue - \{head(queue)\}$;
    if $v.s == d$ then
      $check = 1$; $P \leftarrow v.P$
    else
      for all $v_k$ is neighbour of $v.s$; do
        $node.s \leftarrow v_k$; $node.P \leftarrow P \cup v_k$;
        push $node$ into $queue$;
      end for
    end if
  end while
```

3.4.3 Example Illustration

In this section, we give an example of how our approach works over an arbitrary physical topology G as shown in Figure 3.7(a), with the set of nodes V and edges E.

Since this topology is an unconnected topology, the first step results in a tree T that is a subgraph of G and an unconnected node 13 as in Figure 3.7(b). T is a subgraph of set of nodes V_T and edges E_T, where $V_T = V - \{13\}$, and $E_T = E - \{(2,4),(2,5),(8,9),(11,12)\}$.



(a) An arbitrary topology

(b) The Spanning Tree

(c) Survivable-base result

Figure 3.7: *Example Illustration*

The configuration of the resulting spanning tree of the first step allows us to conclude that G is an unconnected topology. However, further analysis of the physical topology can be performed in the second step, through the the survivable-base algorithm. The input of the second step is the spanning tree T ofFigure 3.7(b), and the output is shown in Figure 3.7(c). Note that topology G contains 3 maximal survivable-bases S_1, S_2, and S_3 and hence the resulted graph S has 3 nodes. Since S_1 and S_2 share nodes 2 in graph G, they are linked by an edge with a weight of 2 (the label of the common node). S_3 does not contain any common node in G with S_1 or S_2, hence S_3 is not connected to other nodes in S.

Combining the results of step 1 and step 2, we can make the following

remarks:

- Graph G is unconnected graph with unconnected node 13.
- Graph G contains three node-survivable subnetworks.
- Graph G contains one link-survivable subnetwork that is $S_1 \cup S_2$.
- Graph G contains at least one link bridge.

3.5 Complexity analysis and numerical results

In this section, we analyse the complexity of our proposed algorithm and evaluate the performance by comparing between this algorithm and cut-set method.

3.5.1 Complexity analysis

The performance of an algorithm can be loosely divided into two majors criteria: 1) a prior estimates, referred as *performance analysis*, and 2) a posterior testing, referred as *performance measurement. Performance analysis* by theory estimates the complexity of an algorithm. The complexity is often evaluated through two metrics: the time complexity and the space complexity. The time complexity of an algorithm is the amount of computer time it needs to run and the space complexity is the amount of memory it needs to run to completion [22]. *Performance measurement* is concerned with obtaining the requirements of time and space of an algorithm. These quantities depend on the compiler used (e.g GNU C++) and the computing platform on which the algorithm is run.

In this part, we theoretically evaluate the performance analysis of our algorithm . The performance analysis is investigated to estimate the complexity of the algorithm and the performance measurement is implemented to make a comparison between our algorithm and the cut-set method. In the next part, the performance measurement is evaluated through a number of simulation and numerical results.

The purposes of the algorithm is to provide the exact configuration of a topology and reduce the time complexity in comparison with cut-set method. Furthermore, since the amount of memory in todays computers is quite large, the space complexity is not as important. Therefore, we only focus on investigating the time complexity of the algorithm. As we mentioned in Section 3.4, our approach is a two-steps process. The time complexity will be analyzed for each step. The complexity is different dependent on the data representation of the network topology such as adjacent matrix and incident link list (ILL).

1. *Time complexity of the connectivity step*:

   The connectivity of a network topology is investigated using spanning tree algorithms such as Prim's algorithm and Kruskal's algorithm. However, these algorithms have been modified in order to adapt to the needs of our algorithm. In addition, the time complexity of these algorithms is different. We shall make a comparison between the original and modified algorithms. The evaluation is performed through two representations of network topology: the adjacency matrix and the incident link list (ILL).

   With respect to adjacency matrix representation, time complexity of original Prim's algorithm is $O(n^2)$ [22]. Although time complexity of the modified Prim's algorithm in the worst case is $O(n^2)$, the average rate of convergence of our modified Prim's algorithm to the solution is, in most cases, much faster than the original because we are not searching for the minimum cost of added edges and hence there may be more than one node added into the tree for each iteration. On the other hand, time complexity of original Kruskal's algorithm is $O(|E|\log|E|)$ while in our algorithm it is $O(|E|)$.

   With respect to the incident link list representation, the time complexity of original Prim's algorithm is $O((n+|E|)\log n)$ [22] while those of our modified Prim's algorithm, in the worst case, is $O(n)$. This is because the search time for minimum cost is not taken into account in the modified Prim's algorithm. The time complexity of both Kruskal's and modified Kruskal's algorithms are not changed in the incident link list representation, compared with the adjacency matrix representation.

   The time complexity of the connectivity algorithms is summarized in Table 3.4, in which STA denotes Spanning Tree Algorithms.

Table 3.4: *Time complexity of the spanning tree algorithms*

		STA		Modified STA	
		Prim	Kruskal	Prim	Kruskal
Time complexity	Adj. matrix	$O(n^2)$	$O(\lvert E\rvert\log\lvert E\rvert)$	$O(n^2)$	$O(\lvert E\rvert)$
	ILL	$O((n+\lvert E\rvert)\log n)$	$O(\lvert E\rvert\log\lvert E\rvert)$	$O(n)$	$O(\lvert E\rvert)$

2. *Survivable-base algorithm*

The time complexity of the survivable-base algorithm is evaluated through two procedures.The first procedure finds the cycles and constructs the network S. The time complexity of this procedure is equivalent to time complexity of BFS searching technique. If G is represented by its adjacency matrix, then the time complexity is $\Theta(n^2)$. The time complexity for T when represented by its incident link list is $\Theta(n + |E|)$. However, if the algorithm is applied over a tree, then the time complexity is $\Theta(2n)$. Since the total number of cycles need to be found is $|E| - n$, the time complexity is $\Theta((|E| - n)(n + |E|))$ for adjacency matrix representation or $\Theta((|E| - n)2n)$ for incident link list. In addition, the time complexity for construct graph S is $O((|E| - n)^2)$. In summary, the time complexity of the first procedure is $O((|E| + n)(|E| - n))$.

The second procedure combines the survivable bases into a larger survivable base through the graph S. This is again is a process of finding spanning tree and cycles, but it is simpler. This is because we do not need to construct a new graph as in the first procedure. The time complexity in this procedure is evaluated similar to the connectivity step and finding cycles procedures.

3.5.2 The performance measurement and numerical results

The performance measurement evaluates algorithms through experiments or simulations. The input data of experiments is often collected in practice. In contrast, the simulation is performed through data which is generated randomly or purposely. Generating a data set that results in the worst-case performance of an algorithm is not always easy. One approach attempt to analyze the algorithm and use a computer to generate the worst-case data set. This approach is difficult in cases of large and complex processes. Another approach is to generate a suitably larger number of random data set. The maximum run time of these data sets is used to estimate the worst case time.

On the other hand, estimating the average time of an algorithm is not usually possible. It is possible in some cases such as sequential search, but it is not possible for a sort algorithm. Similarly, it is possible for our algorithm, but it is not possible for the cut-set method. In graphs of the same nodes, the time complexity of cut-set algorithms is independent of the number of links while those in our algorithm depends on the number of network links and the topology of the networks.

In the simulation, we attempt to examine the computational efficiency and the identification to survivable topology in comparison between our proposed algorithm and the cut-set method. Since the cut-set algorithm is intractable with large scale networks, our simulation generates two set of data. The first data set is generated for small networks and the second is for large scale networks. The comparison between our algorithm and cut-set algorithm is performed over the first data set. The second data ̦set evaluates the enhancement of our algorithm with large scale networks. Details of these data sets are as follows:

(a) *Data set 1:*

Random networks are generated with the number of nodes from 10 to 20. Corresponding with a number of network nodes, we generate 1000 random network topologies that contains an average nodal degree of $\{2, 2.5, 3, 3.5, 4\}$. For examples, there are 1000 topologies of 10 nodes with an average nodal degree of 2 and, 1000 topologies of 10 nodes with an average nodal degree of 2.5, and so on.

(b) *Data set 2:*

This has a similar network configuration to data set 1 but the number of network nodes generated are varied from 100 nodes to 500 nodes.

We simulate and compare the results of our proposed algorithms and the cut-set method. The results are compared and analyzed according to two factors: the solution accuracy and the computational time as below.

- **The solution accuracy:**

  Table 3.5 shows the results of the cut-set method and the survivable-base algorithm. These are implemented and measured through the same set of random physical topologies.

  For clarity, we denote notations used in these tables as follows:

Notations	Meaning
N	the number of network nodes.
Degree	the *average nodal degree* of the network.
UCT	the number of *unconnected topologies.*
UST	the number of *unsurvivable topologies.*
ST	the number of *survivable topologies.*
LST	the number of *link-survivable topologies.*
NST	the number of *node-survivable topologies.*

  It can easily seen that the cut-set method only yields if a topology is survivable. The column **ST** in Table 3.5 shows the number of survivable topologies out of 1000 random topologies verified. We note that a survivable topology which is returned from the cut-set method may be either a link-survivable or a node-survivable. However, the cut-set technique can not identify them. In other words, the results of the cut-set method can only assure that if a topology can support link-survivable, but not node-survivable.

The proposed survivable-base algorithm has resolved the problem of cut-set. The class of a physical topology can be identified using the algorithm. The **LST** and **NST** columns in Table 3.5 show the number of link-survivable and node-survivable topologies out of 1000 random verified topologies. Note that **LST** denotes the number of topologies which are link-survivable, not node-survivable. Hence, the total number of link-survivable (**LST**) and node-survivable (**NST**) topologies returned from the survivable-base algorithm is equal to the number of survivable topologies (**ST**) returned from the cut-set method. In addition, as demonstrated in Part Section 3.4.3, the survivable-base algorithm can further point out the weaknesses of a topology such as node-brides or link-bridges, whose failures disconnect the topology. Node-bridges and link-bridges is obtained using the survivable-base algorithm in this simulation. However, with the data set of 1000 topologies for each circumstance of the networks, the presentation of these results is massive and not significant.

- **The computational time**

  The computational time of cut-set method is independent with the number of network links, that is, the cut-set achieves the same computational time for every topology configuration of networks that contains the same number of network nodes. In addition, the computational time is intractable with large scale networks because of the explosion of the number of cut-sets. Figure 3.8 shows the trends of the computational time. This is an exponential curve according to the increasing of network nodes. For example, the computational time for networks of 10 nodes on average is around 0.5 seconds, this number for 15 nodes and 20 nodes is around 14 seconds and 506 seconds, respectively, that is, the computational time increases around 32 times (2^5) for each 5 nodes increased of network nodes. This is reasonable because the number of cut-sets, by theory, is $2^N - 2$, which is a exponential function of the number of network nodes N. It is no doubt to note that the cut-set method is not applicable for even with moderate scale networks. The estimation of verification time of the cut-set method for a network of 25 nodes is about 4 hours and 30 minutes, for a network of 30 nodes is about 6 days

Table 3.5: *The comparison results of the survivable-base algorithm and the cut-set method for topology identification.*

N	Degree	Topologies	Cut-set method			Survivable-base approach			
			UCT	UST	ST	NC	NS	LST	NST
10	2	1000	744	256	0	744	256	0	0
	2.5	1000	300	570	130	300	570	88	42
	3	1000	166	588	246	166	588	44	202
	3.5	1000	49	410	541	49	410	6	535
	4	1000	19	222	759	19	222	2	757
15	2	1000	933	67	0	933	67	0	0
	2.5	1000	606	345	49	606	345	48	1
	3	1000	365	548	87	365	548	40	47
	3.5	1000	205	581	214	205	581	21	193
	4	1000	82	473	409	82	473	5	440
20	2	1000	987	13	0	987	13	0	0
	2.5	1000	798	171	31	798	171	31	0
	3	1000	517	453	30	517	453	22	8
	3.5	1000	298	588	114	298	588	16	98
	4	1000	161	601	238	161	601	5	233

and for a network of 40 nodes is about one year[1].

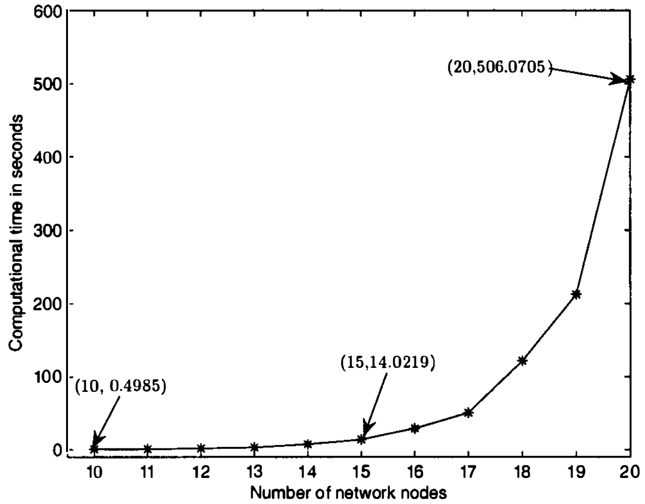


Figure 3.8: *Computational time - cut-set method*

The survivable-base algorithm can resolve the problem of the cut-set method. The proposed algorithm can cope with a large scale networks. For the comparison between cut-set method and the survivable-base algorithm, we implement the algorithm in the same scenario of the network with cut-set. The result, shown in Figure 3.9, shows the computational time of the survivable algorithm according to the number of network nodes. For each number of network nodes, the computational time depends on the number of links in the networks. In the simulation, we examine the computational time of the algorithm according to the variety of network links represented by the average of nodal degree. For example, the average of links in the network corresponding to the average of nodal degree 2 is 2 × N, where N is the number of network nodes.

It can be easily seen that, survivability checking of the survivable-base algorithm is much faster than the cut-set method; and the gap of the computational time between these algorithms becomes larger with the increase in the number of network nodes. The computation time of survivable-base algorithm for network of 10 nodes in the worst case is around 0.05 seconds, which is much faster than the cut-set method (around 0.5 seconds). The computational time in the worst case of survivable-base algorithm is still less than 0.3 seconds, compared to 506 seconds in the cut-set method.

In order to illustrate the advantage of the survivable-base algorithm, we

[1] Run on Pentium 4, 1.8Ghz and 384Mb of Ram

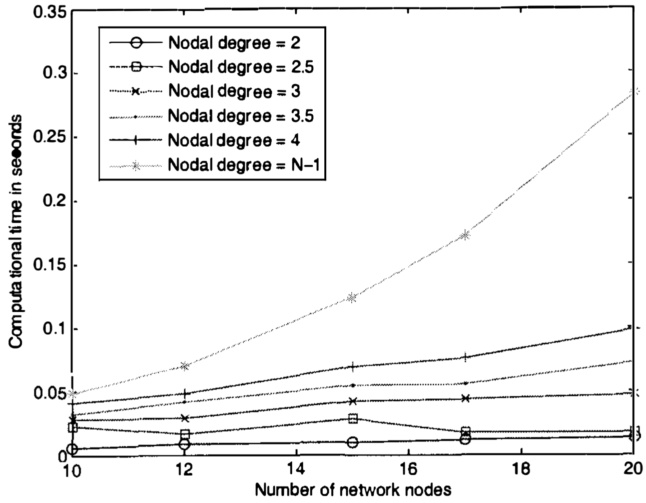
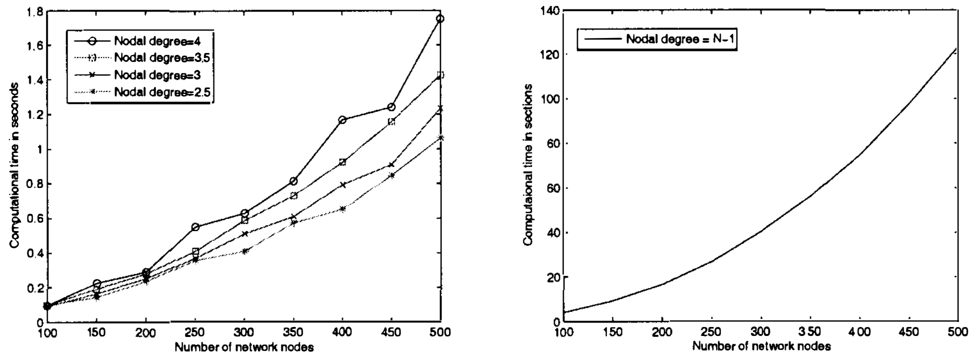


Figure 3.9: *Computational time of survivable-base*

check for survivability in large scale networks, where the number of network nodes is varied from 100 to 500 nodes. The result is shown in Figure 3.10.



(a) *Time computation in large scale networks*

(b) *The worst-case computational time for fully connected mesh networks*

Figure 3.10: *The computation time of the survivable-base algorithm*

We can see that the performance of survivable-base algorithm in the sparse networks is excellent in terms of the time complexity. The computational time for networks with average nodal degrees in the range from 2.5 to 4 is very low. For example, the value for network of 100 nodes and nodal degree 4 is around 0.1 seconds and is increased to less than 1.8 seconds for network 500 nodes. This value does not increase significantly even for the sparse networks of thousands nodes.

Although the computational time of survivable-base algorithm is higher in the worst-case of large scale networks (just over 120 seconds for complete networks of 500 nodes), the algorithm still offers an acceptable computational time, even with networks of thousands nodes .

3.6 Concluding Remarks

The proposed approach has resolved two problems of cut-set technique for survivability at physical layer. First, the implementation of our physical survivability framework can clearly identify the weaknesses of a network. For the first time, it is possible to establish the survivability of the network not only on the basis of link failures, but also with respect to node failures. Furthermore, our solution gives a comprehensive diagnosis of the network and identifies the exact nodes and links which are the weaknesses of the network, making it unsurvivable. This cannot be done using the cut-set method which can only assure if a topology is link-survivable. Secondly, the survivability of the logical topology is heavily dependent on the survivability of the physical topology, establishing the physical survivability of the network is of utmost importance. However, the existing techniques were not able to establish the physical survivability of a moderate size network in a reasonable amount of time. For instance, as we stated in the simulation, the cut-set technique is not applicable to a network which has more than 30 nodes. In this thesis, we provided a novel theoretical framework for the assessment of the physical survivability of the network. Our proposed algorithms based the framework can cope with large size networks, even in the order of many thousand nodes. For instance, the computational time of the algorithm for a sparse network of 100 nodes is only around 0.1 seconds and increases to less than 1.8 seconds for a network of 500 nodes. These values are measured for sparse networks in which the average nodal degree is ranges from 2 to 4. In the worst-case, the computational time for fully connected mesh networks of 100 to 500 nodes is still acceptable and remains in the order of seconds.

Chapter 4

Survivable Logical Topology Design

Above the physical topology, traffic connections are provisioned as lightpaths, termed as *logical connections.* In addition, when the connections require to be protected against failures, the logical connections must include two routes: the working route and the backup route. The working route is employed to carry data in normal operations and the backup route is used to protect against network failures. We refer to such logical connections as *survivable logical connections* and the process of establishment such connections as *Survivable Logical Topology Design* (SLTD). This is an optimization problem in which traffic connections are routed subject to specific objectives according to requirements of service providers, that may be network cost or network performance (blocking, congestion). Therefore, SLTD is one of crucial issues, beside physical topology design, that service providers concern to utilize available network resources and network performance.

Network performance, with respect to network survivability, is measured through several metrics such as the restoration time and the restoration probability that we refer as restorability. These differ from protection scheme to protection scheme. Research has shown that there is a trade-off between capacity utilization and restorability [11, 1]. Path protection utilizes network capacity more efficiently than link protection, but link protection has a faster restoration time. Similarly, shared protection provides significant capacity savings over dedicated protection, but restoration probability of working routes in shared protection need to be carefully considered.

Network survivability in the design perspective is concerned with two important issues. Firstly, the provisioning of traffic connections, known as Routing Wavelength and Assignment (RWA), which has been proven to be NP-hard [33]. Integer Linear Programming (ILP) formulation can be employed to achieve the exact optimal solution, but it is intractable even with moderate scale networks. Although several heuristic approaches have been proposed to obtain near-optimal solutions, it is a challenge to achieve optimal solutions which are computationally efficient.

Secondly, as an optimization problem, survivable logical topology design is subject to specific objectives that may relate to either network performance or network resource utilization. Amongst network performance measures, minimizing congestion level has been extensively studied in the literature [19, 12]. This can be considered as a load balancing problem in which traffic connections are provisioned so that either the average congestion or the maximum congestion in the network is controlled. On the other hand, with respect to network resource utilization, research community has often focused on capacity efficiency, particularly the number of wavelength channels used in optical networks [11]. It is observed that, for schemes in which the maximum network congestion level is intended to be achieved, the number of wavelength channels used in the solution may be very high. Load balancing aims to reduce the blocking probability for future provisioning, but the network resource may be quickly exhausted. Conversely, in schemes with the second objective, the congestion or capacity used in some links may reach to its limitation although the total wavelength channels used in the network is minimum. Such links are blocked to the next required connections, and in some cases the requests may be refused even though there are still many available wavelength channels. It is, therefore, imperative and crucial to utilize network capacity whilst maintaining a low congestion in the network.

In this chapter, we investigate the-state-of-the-art network survivability at logical topology design. Protection schemes (link/path protection, and shared/dedicated protection) are investigated through popular methods such as Integer Linear Programming formulation and two popular graph approaches, namely the two-step and the one-step approaches. Our objective is to balance the conflicting requirements of network capacity utilization and network congestion level. In addition, as mentioned earlier, SLTD is an NP-hard problem where the optimum solutions are achieved at the extreme cost of the computational time. Our objective is to devise a new heuristic approach that can control the optimality of solutions in acceptable computational time for moderate scale networks.

Our contributions from this chapter are three folds. First, path/link protection and dedicated/shared protection schemes are summarized and implemented through popular approaches in literature, namely ILP formulation and graph theory approaches. We propose a heuristic approach that combines the computational advantages of graph algorithms and optimal solutions of ILP formulation with small number of binary decision variables and constraints. Secondly, an integrated capacity efficiency and network congestion level objectives is proposed. This allows us to control the compromise between network resource utilization and network performance. Finally, a theoretical framework to this objective for protection schemes is developed. This framework provides a general view that includes the following issues: 1) how can an optimization objective be obtained to specific purposes of service providers; 2) how can the dedicated and shared protection schemes be modeled; and 3) how can a route be provisioned to keep the congestion level as low as possible.

The rest of this chapter is organized as follows. Details of the problem in our study is stated in Section 4.1 and Section 4.2. A theoretical framework of our work to survivable logical topology design is presented in Section 4.3. Next, in Section 4.4, we investigate two existing approaches and propose a new heuristic approach that combines the advantages of solutions accuracy in ILP model and computational time efficiency in graph algorithms. Finally, numerical results that compare the efficiency of our approach over existing ILP formulation and graph algorithms are presented in Section 4.5

4.1 Problem Formulation

For clarity of our discussion, we define two objective metrics as follows:

- *The number of wavelength channels used:* this is the total number of wavelength channels assigned in all network links. In this chapter, we refer a wavelength channel as a unit of network capacity.
- *Congestion level:* the congestion index of a link is the capacity used in that link. Normally, congestion level is defined as the maximum link congestion index, and an average of the link congestion index is called the average congestion level.

Given a physical optical network and static traffic requirements, SLTD's aim is to map the connections over the physical topology to carry required traffic in the network. The embedding process must comply with survivability conditions and optimization purpose. In our work, the performance of approaches to SLTD problem is investigated and compared through the following criteria:

1. Capacity utilization
2. Network congestion

3. Time complexity

4. Optimality of solutions

The performance of SLTD, however, depends on various factors of network environment such as physical topology, traffic pattern and survivability requirements. For consistency, network environment assumptions have to be identified.

4.1.1 Network Assumptions

There are a number of assumptions that we make about physical netwok configuration, traffic requirements and protection requirements. These are discussed below.

Physical network configuration

A physical topology of an optical networks is given. This physical topology have the following characteristics:

- *Network nodes:*

  Network nodes are assumed to contain Optical Cross Connects (OXCs) and have full wavelength conversion capability. This allows us to focus only on the problem of wavelength routing. In addition, data can be added/dropped in each node by optical add/drop multiplexers (OADM).

- *Network spans and network links:*

  Network spans are optical fibers connecting two nodes in the networks. In practice, a span may contain one or more fiber cables. One fiber cable in the span is defined as a network link. A span in general is, therefore, a set of network links. In this work, we assume that there is only one optical fiber in each span, and hence we refer to network spans as network links.

  Network links are bidirectional. Data flows can be carried from one end-node of a link to another and vice versa. The assumption also implies that all wavelength channels on network links are bidirectional. However, if a wavelength channel is used in one direction, then it cannot be used to allocate for any connection in the reversed direction. Another assumption for network links is that the number of wavelength channels provided by network links is the same.

- The topology must be able to offer link disjoint or node-disjoint path-pair between any two nodes. In other words, this topology is link-survivable or node-survivable as defined in Chapter 3. The check for survivability at physical layer can be done through the approach proposed in the previous chapter.

Traffic requirements

- Traffic requirements are given in advance. These requirements are usually estimated as long-term traffic.
- All traffic requirements are directional, that is, the requirement from one node to another differs from those in the reversed direction.
- Since a request between any two nodes in practice is random, we assume that traffic connections are generated randomly and the probability distribution of connection request between any two nodes in the network is homogeneous

Protection requirements

- The impact of failures on network performance is measured by network restorability. In our study, network restorability is assumed to be one hundred percent. In other words, under network failures, SLTD has to setup backup paths so that all affected working channels are restored.
- In practice, multiple failures may occur simultaneously, but it is very rare. Therefore, all protection schemes in this work are designed to cope only with single link failures.
- Protection schemes are examined over link protection and path protection. In each scheme, dedicated protection and shared protection are investigated.

4.1.2 Problem formulation summary

Given the network assumptions, the SLTD can be summarized as follows.

Input parameters

- **The physical topology:**

  A bidirectional mesh topology is given as graph $G(V, E)$, where V is a set of N network nodes and E is a set of M network links.

- **The traffic requirements:**

  Required unidirectional connections are given as $T = [t^{sd}]$, $s, d \in \{1..N\}$

- **The wavelength limit:**

  W is the number of distinct wavelength channels on each fiber link.

SLTD process

- We investigate the network performance through an integrated objective function network capacity used and network congestion level.
- Network survivability schemes, including link/path protection and dedicated/shared protection, are examined using three approaches, namely ILP formulation, graph algorithms and our proposed approach. These schemes are implemented to comply with the above assumptions.

Output parameters

- **Routing tables:** the routing tables include working routes table and backup routes table.
- **An optimized objective value:** the value contains information about utilized network capacity and congestion level in the network.

4.2 Problem Setting

Survivable Logical Topology Design is quite complex, both in concept and implementation. This is because of the fact that SLTD problem involves many mathematical fields such as graph theory and linear algebra. Network models and notation, thereby, differ from one approach to another. For consistency in our discussions, our network model and notation are defined for the three approaches: ILP formulation, graph algorithms, and our proposed approach.

4.2.1 Network model

Network model includes two types of models, namely the topology model and the cost model. In the topology model, the structure of a physical network can, in general, be modeled as a graph that contains the whole information about this network, eg. network node label or network link direction. The cost model, on the other hand, represents the economic value of network components.

Topology model

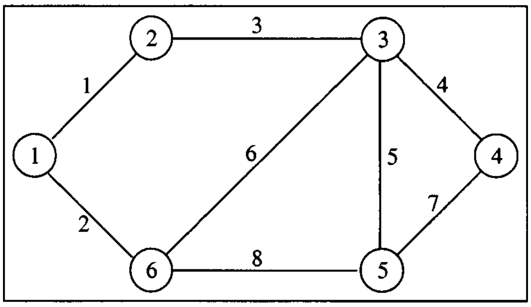


Figure 4.1: *An example of physical network topology*

A physical network is modeled as a graph, an example is shown in Figure 4.1. Since the links of the physical topology are assumed to be bidirectional, the graph is undirected. Our mathematical notation in this model is as follows:

- *Network nodes* are represented as a set $V = \{v_1, v_2, \ldots, v_N\}$, where N is the number of network nodes. For convenience of implementation, a node v_k is referred to as node k. Therefore, terms "node v_k" and "node k" have the same meaning.
- *Network links*
  - Network links are represented as a set $E = \{e_1, e_2, \ldots, e_M\}$, where M is the number of network links.
  - A link e_i is numbered according to the order of network nodes and enumerated as follows: link e_1 connects nodes v_1 to node v_k of its neighbours where k is the lowest index amongst all neighbours; link e_2 connects nodes v_1 and the next lowest index nodes in its neighbours. The enumerating for remaining links follows the same rule. For example, in Figure 4.1 link e_1 connects nodes v_1 and v_2, link e_2 connects nodes v_1 and node v_6, and so on.
  - We also take the terms "link e_k" and "link k" to have the same meaning.

With respect to implementation, a graph G is represented through an adjacency matrix A; and the link capacity of the graph is represented as a link list LL. These are defined as follows:

- *The adjacency matrix:*

  The adjacency matrix A is a $N \times N$ matrix, where N is the number of network nodes. This matrix is represented in Equation 4.1 as below.

$$A = \begin{pmatrix} a_{11} & a_{i2} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix} \tag{4.1}$$

In this representation, element a_{ij} shows the connection between node v_i and node v_j, and is given by:

$$a_{ij} = \begin{cases} 0 & \text{if } v_i \equiv v_j \\ c_{ij} & \text{if node } v_i \text{ connects to node } v_k \\ \text{inf} & \text{otherwise} \end{cases}$$

where c_{ij} is the cost of the link connecting nodes v_i and v_j, and it is discussed in the next section (the cost model). If there is no link between nodes v_i and v_h, then its cost is set to be infinity (inf).

In undirected graphs, the adjacency matrix A is symmetric or $a_{ij} = a_{ji}$. For example, the adjacency matrix for the example graph in Figure 4.1 is given by:

$$A_E = \begin{pmatrix} 0 & 1 & - & - & - & 1 \\ 1 & 0 & 1 & - & - & - \\ - & 1 & 0 & 1 & 1 & 1 \\ - & - & 1 & 0 & 1 & - \\ - & - & 1 & 1 & 0 & 1 \\ 1 & - & 1 & - & 1 & 0 \end{pmatrix} \tag{4.2}$$

where symbol '$-$' denotes for inf and we have assumed that all link costs are one unit.

- *The link list LL*

A matrix of $M \times 2$ is employed to represent the link list LL as:

$$LL = \begin{pmatrix} 1 & W_1 \\ 2 & W_2 \\ \dots & \dots \\ M & W_M \end{pmatrix}$$

The first column of each row shows the order of links in the graph and the second column shows the number of capacity units provided on those links. In our study, one capacity unit on a link is one wavelength channel on that link. In the example graph, we have assumed the capacity provided on all network links to be equal to 12, and hence the LL matrix is given by:

$$LL_E = \begin{pmatrix} 1 & 12 \\ 2 & 12 \\ \dots & \dots \\ 8 & 12 \end{pmatrix}$$

The Cost Model

The cost of a physical network is the total cost of its components, namely network nodes and links. An optical node, in general case, contains numerous devices such as OXCs, OADMs and OLTs, and hence the cost of a node is the total cost of devices contained in it. Meanwhile, network links are optical cables, also known as fiber links, and hence link cost is the cost of fiber links. The cost of a link normally is calculated through the distance of the link and the cost of fiber links.

In our study, network cost is considered in a different way. Since our objective is to minimize utilized network capacity and reduce maximum congestion level, we mainly focus on how many capacity units in a link have been allocated. A unit of cost in the network is defined as one wavelength channel on one link. Therefore, if w wavelength channels are used on link e_k, the total usage cost of that link is w. According to this definition, the cost of a route in the network is equal to the number of links contained in the route.

In summary, the network cost in our study is modeled as follows:

- If link e_i is provided with W_i wavelength channels, the cost of this link is W_i
- The network cost is the total cost of network links.
- Path (route) cost is the number of links contained in the path (route).

4.2.2 Notation

The following notations are adapted for 1) network structure; 2) ILP formulation; and 3) graph theory.

Network notation

- A physical topology is modeled as an undirected graph $G(V, E)$, where V is a set of network nodes and E is a set of fiber links.
- Network nodes in V is enumerated as $\{v_1, v_2, \ldots, v_N\}$, where N is the number of network nodes.
- Network link in E is enumerated as $\{e_1, e_2, \ldots, e_M\}$, where M is the number of network links.
- Since a link e_j is bidirectional, the link which connects nodes v_l and v_k is be denoted by $e_j = [v_l v_k]$ or $e_j = [v_k v_l]$. However, for convenience, where $v_l < v_k$, we denote this link as $e_j = [v_l v_k]$.
- $IE_k \in E$ is set of incident links of node v_k.
- W_i denotes the number of wavelength channels provided on link e_i. When W_i is the same for all $i = 1 \ldots M$, we simply denote it as W.
- Let T denote the traffic requirements. T is a set of traffic connections as $T = \{t_i(s, d) | i = 1 \ldots D, (s, d) \in [1 \ldots N]\}$, where i is the index of connections, s and d are source node and destination node of connection t_i, and D is the number of connections required.

ILP formulation notation

The purpose of our study is to find the working route and the backup route of all connection requests from source nodes to destination nodes. Since these routes are directed while network links are undirected, direction of links to indicate the flow of the routes need to be defined. The definition is as follows: *forward direction* of a link is started from the smaller order end-node to the larger end-node, and *backward direction* is started from larger order end-node to the smaller. For example, link 3 of graph in Figure 4.2 connects nodes 2 and 3, hence the forward direction of link 3 is from node 2 to node 3 and the backward direction is from node 3 to node 2.

In the SLTD problem, working routes and backup routes have to be determined. However, a route may be allocated in arbitrary links with arbitrary direction. For that reason, in order to identify whether a route traverses through a link or not, two binary decision variables are employed. One variable indicates if a route traverses through a link in forward direction; and another variable indicates it for backward direction. For example, let p be a required route from node 1 to node 4 as in Figure 4.2. We denote

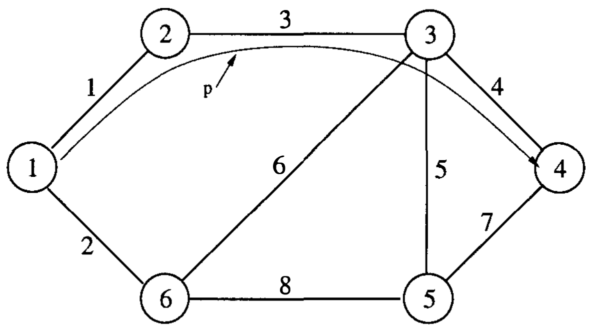


Figure 4.2: *An example of physical topology modeled in ILP*

binary variable p_{+i} (p_{-i}) to indicate whether route p traverses though link e_i in forward (backward) direction. In this example, p_{+1}, p_{+3} and p_{+4} are 1 and other variables are equal to 0. We further notate:

- w^i_{+j} is 1 if primary path of connection t_i uses link e_j in forward direction, otherwise w^i_{+j} is 0.
- w^i_{-j} is 1 if primary path of connection t_i uses link e_j in backward direction, otherwise w^i_{-j} is 0.
- r^i_{+j} is 1 if backup path of connection t_i uses link e_j in forward direction, otherwise r^i_{+j} is 0.
- r^i_{-j} is 1 if backup path of connection t_i uses link e_j in backward direction, otherwise r^i_{-j} is 0.
- r^i_{+j+l} is 1 if forward direction of link e_l is used for backup path of working channel in forward direction of link e_j of connection t_i, otherwise r^i_{+j+l} is 0.
- r^i_{+j-l} is 1 if backward direction of link e_l is used for backup path of working channel in forward direction of link e_j of connection t_i, otherwise r^i_{+j-l} is 0.
- r^i_{-j+l} is 1 if forward direction of link e_l is used for backup path of working channel in backward direction of link e_j of connection t_i, otherwise r^i_{-j+l} is 0.
- r^i_{-j-r} is 1 if backward direction of link e_l is used for backup path of working channel in backward direction of link e_j of connection t_i, otherwise r^i_{-j-l} is 0.
- f_{max} is maximum congestion index on fiber links, and is referred as congestion level.
- f_{sum} is the total unit capacity used in the network.
- $\lceil x \rceil$ is nearest integer number which is larger than or equal to x.

- $\Sigma(x_{\pm i}) = x_{+i} + x_{-i}$.

- $x_{\pm i}$ means x_{+i} or x_{-i}.

- $\Sigma(x_{\pm i \pm j}) = x_{+i} + x_{-i} + x_{+j} + x_{-j}$.

- $x_{\pm i \pm j}$ means x_{+i+j} or x_{-i+j} or x_{-i+j} or x_{-i-j}.

Note that since we investigate network survivability in two scenarios of path protection and link protection, the notation for backup routes is different between these protection scenarios.

Graph notation

In the graph theory approach, traffic requirements are provisioned for connections sequentially, and network capacity usage and congestion are increased accordingly. For optimization purpose, network status is considered in terms of capacity efficiency and congestion. We introduce the following notations to indicate the status of the network before and after a connection is provisioned:

- p^j is a candidate route for connection t_j.

- $P^j = \{p^j_1, p^j_2, \ldots, p^j_K\}$ is a set of K candidate routes for connection t_j.

- c^j_i denotes the total unit capacity used in route p^j_i.

- f_i denotes the number unit capacity used in link e_i.

- f_{max} is the maximum congestion amongst f_i.

- $f^{c,j}_i$ denotes the number unit capacity used in link e_i before connection t_j is provisioned.

- $f^{c,j}_{max}$ is the maximum congestion before connection t_j is provisioned.

- $f^{p,j}_i$ denotes the number unit capacity used in link e_i after connection t_j is provisioned.

- $f^{p,j}_{max}$ is the maximum congestion after connection t_j is provisioned.

4.3 Survivable Logical Topology Design: Theoretical Analysis

In this section, we discuss a theoretical framework to the problem of survivable logical topology. Firstly, we discuss our objective for optimization purposes, and explain how our work can balance the conflict between capacity efficiency and network congestion level. Next, we investigate the allocation of wavelength channels for working routes and backup routes. Finally, a principle for provisioning traffic connections is proposed. This principle is only applied in graph algorithms.

4.3.1 Optimization objective

As mentioned earlier in this chapter, minimizing the number of wavelength used and minimizing the maximum congestion level are two common objectives used for optimization of the network. With respect to the first objective, optimal capacity can be performed, but network congestion level may be very high. Conversely, when network congestion level is minimized, network capacity may be quickly exhausted. Our aim is to minimize the number of network capacity units used whilst maintaining a low congestion.

- Let f_i be the total capacity used in link e_i and referred to as *link congestion index*.
- Let f_{sum} be the total capacity used in the network, then f_{sum} is determined as:

$$f_{sum} = \sum_{e_i \in E} f_i \tag{4.3}$$

  f_{sum} varies in the range from 0 to the total capacity provided in the network. In other words, $0 \leq f_{sum} \leq MW$, where M is the number of network links and W is the number of wavelength channels provided on each link.

- Let f_{max} be maximum congestion in the network, then f_{max} is determined as:

$$f_{max} = \max_{e_i \in E} f_i \tag{4.4}$$

  The range of f_{max} is: $0 \leq f_{max} \leq W$.

Given a physical topology and a set of traffic connections, we observe intrinsic properties when provisioning the connections over the network as follows.

- There may be more than one minimal solutions of capacity utilization. In other words, there may be different routes for connections which achieve the same minimum value of f_{sum}. It is a challenge to select the solution with the lowest congestion.

- Similarly, there may be different routes to achieve the same minimum value of f_{max} and the challenge is to chose the solution that has the smallest number of unit capacity used.

In our work, we propose general mathematical formula that can be controlled to strike a compromise between the two conflicting objectives. Let consider the expression.

$$f_{com} = Af_{sum} + Bf_{max} \tag{4.5}$$

where we defined f_{com} as an integrated objective of f_{sum} and f_{max}, and A and B are two constants.

The objective now is to minimize f_{com} instead of f_{sum} or f_{max}. We investigate the dependence of f_{com} on f_{sum} and f_{max} through different values of the constants A and B.

1. **For $B = k$ and $A > kW$:**

   Let $A = kA', \Rightarrow A' > W$.

   The objective function f_{com} is given as:

   $$\begin{aligned} f_{com} &= kA'f_{sum} + kf_{max} \\ &= k(A'f_{sum} + f_{max}) \end{aligned}$$

   We assume that f_{sum} increases by 1 or $f'_{sum} = f_{sum} + 1$ and f_{max} varies in its limited range (known as f'_{max}), then the new value for f_{com} is f'_{com} and calculated as:

   $$\begin{aligned} f'_{com} &= k(A'f'_{sum} + f'_{max}) \\ &= k(A'(f_{sum} + 1) + f'_{max}) \\ &= k(A'f_{sum} + f'_{max} + A') \end{aligned}$$

   Therefore,

   $$\begin{aligned} f'_{com} - f_{com} &= k\left(A'f_{sum} + f'_{max} + A' - (A'f_{sum} + f_{max})\right) \\ &= k\left(A' - (f_{max} - f'max)\right) \end{aligned}$$

$$0 \leq f_{max}, f'_{max} \leq W \quad \Rightarrow \quad -W \leq (f_{max} - f'_{max}) \leq W$$

$$\therefore \quad (f'_{com} - f_{com}) > 0 \text{ or } f'_{com} > f_{com} \tag{4.6}$$

Equation 4.6 shows that f_{com} is changed significantly even with a small change in f_{sum}. In this case, corresponding to each unit increasing in f_{sum}, f_{com} increases by a factor of A. Thus, f_{com} is still increasing even when f_{max} decreases by the maximum value (W).

Therefore, we conclude that in order to minimize f_{com}i, we first minimize f_{sum} and then f_{max}.

2. **For $A = k$ and $B > kMW$:**

   Let $B = kB', \Rightarrow B' > MW$.

   The objective function f_{com} is given as:

$$\begin{aligned} f_{com} &= kf_{sum} + kB'f_{max} \\ &= k(f_{sum} + B'f_{max}) \end{aligned}$$

   Similar to the first condition, we examine the case when f_{max} is increased by 1, and f_{sum} varies in its range (represented as f'_{sum}). We have,

$$f'_{com} - f_{com} = B' - (f_{sum} - f'_{sum})$$

$$0 \leq f_{sum}, f'_{sum} \leq MW \quad \Rightarrow \quad -MW \leq (f_{sum} - f'_{sum}) \leq MW$$

$$\therefore \quad (f'_{com} - f_{com}) > 0 \text{ or } f'_{com} > f_{com} \tag{4.7}$$

   Therefore, in order to minimize f_{com}, we first minimize f_{max} and then minimize f_{sum}.

In summary, f_{com} can be represented as:

$$f_{com} = f_{sum} + \alpha f_{max} \tag{4.8}$$

where α is called *the weighting factor.*

If $\alpha < \frac{1}{W}$, then the first priority is to minimize the total capacity usage. Meanwhile, if $\alpha > MW$, then network congestion level is minimized first[1].

[1] Discussion of the range for α from $\frac{1}{W}$ to MW is beyond the scope of this thesis

4.3.2 Capacity allocation

In our study, network survivability is investigated through path protection and link protection scenarios. In each scenario, dedicated and shared schemes are considered. The capacity allocation for working paths and backup paths is not different between dedicated and shared schemes. However, network capacity units assigned to backup paths differs between these two schemes. In this part of work, we investigate the mechanism of network capacity allocation to working and backup paths for dedicated and shared schemes. With respect to working routes, if there are w working routes traversing through link e_i, then the number of capacity units assigned to these paths is $w_i = w$. This allocation is the same for dedicated and shared protection schemes. With respect to backup paths, we consider dedicated and shared schemes as follows.

Dedicated scheme

In dedicated protection scheme, backup paths are dedicated for their working paths, hence wavelength channels are assigned to backup paths separately. In other words, backup paths of connections do not use the same wavelength channels. Therefore, if there are r backup paths traversing through link e_i, then the total capacity allocated to these backup paths is $r_i = r$. The total capacity used in link e_i is given as:

$$f_i = w_i + r_i$$

where w_i is the number capacity allocated to working paths.

Shared scheme

The difference between dedicated and shared schemes is in the way that the network capacity is assigned to backup paths. In shared protection schemes, a backup path of a connection t_j in link e_i is assigned either a new wavelength channel or a shared channel. This is identified according to a so-called *shared factor* C_s:

$$C_s = \frac{P}{Q}$$

where P is the number of shared channels and Q is the number of required channels $(Q \geq P)$.

Therefore, if there are r backup paths required to be allocated in a link, then the number of wavelength channels used is:

$$r_{used} = r \times C_s$$

or,

$$r_{used} = r \times \frac{P}{Q} \tag{4.9}$$

Since the number of wavelength channels is an integer, r_{used} must be an integer number. However, C_s is a real number and $0 \leq C_s \leq 1$. Hence r_{used} in Equation 4.9, in general, is a real number. We modify Equation 4.9 to ensure that r_{used} is an integer number while it still satisfies the shared condition:

$$r_{used} = \lceil r \times \frac{P}{Q} \rceil \tag{4.10}$$

r_{used} in Equation 4.10 is a smallest integer number larger than or equal to r_{used} in Equation 4.9.

If the number of backup paths in a link equal to Q ($r = Q$), then the number of wavelength channels used in the link for backup paths is P. In other words, every Q backup path can share every P available wavelength channels in a link. Equation 4.10 will be used to allocate backup paths throughout our work in this chapter.

4.3.3 Route provisioning

This part of our work is developed as a framework for provisioning traffic connections in a graph theory approach. Since connections in this approach is provisioned sequentially, at the time of making every provision, the total network capacity used must be minimized and the network congestion level need to maintained as low as possible.

1. *Minimize total capacity*

   A minimal feasible solution in provisioning of a network connection is a route whose cost is minimum. In fact, this is a least cost path between end-nodes of the connection and can be identified through existing shortest path algorithms such as Dijkstra's algorithm or Bellman-Ford's algorithm. However, there may be more than one such least cost route between two nodes in the network, amongst which we need to select one to provision for the connection. In our work, minimizing congestion level is used as a criterion selecting a such route. Obviously, regardless of which candidate minimal cost routes is selected, the total capacity used is always minimum.

   With respect to our objective, the task is to find as many minimal feasible solutions as possible. However, to our best knowledge, there is no mathematical theory to determine exactly the number of paths whose costs are equivalent and minimum. The task, however, can be done by computer algorithms, namely K shortest paths algorithms. These algorithms have usually been implemented in the literature [34, 35, 36] with a given constant K. As a result, the first K shortest paths found are yielded. Our problem, however, is different. We attempt to find the first K shortest paths whose costs are equivalent to each other. In other words, the value of K is different from one connection to another connection. We therefore adopt and modify the K shortest paths algorithms to serve our purpose. A counting variable is introduced to monitor the number of shortest paths found. The algorithm is terminated when the cost of the next shortest path found is larger than the previous paths.

2. *Minimize network congestion level*

   Let F be a discrete distribution of f^i, where f^i is the number of wavelength channels used in link e_i, then F is a set of link congestion indexes.

   Network congestion f_{max} is defined as the maximal congestion of all network links. In other words, network congestion level can be mathematically defined as:

$$f_{max} = \max_{e_i \in E}(f^i)$$

   The average congestion in the network is the average of f^i and is represented as:

$$\text{mean}(F) = \mu = \frac{1}{M}\sum_{i=1}^{M} f^i$$

   The distribution of f^i around their average value is measured through the variance or standard deviation:

$$\text{var}(F) = (\sigma)^2 = \frac{1}{M}\sum_{i=1}^{M}(f^i - \mu)^2$$

   Our objective for provisioning connections is to minimize the influence of provisioned route on the current congestion level. The influence is investigated through three factors: *mean μ of the distribution F, the increment of f_{max}* and *variance* σ^2. For convenience of discussion, notation of status of networks before and after a route is established is defined as follows:

   Let $f^{c,i}$ be link congestion index of e_i before a connection T_j is provisioned and p_j be a route of T_j, then the link congestion index $f^{p,i}$ of e_i and F^p is determined as: rk

$$f^{p,i} = \begin{cases} f^{c,i} + 1 & \text{if } p_j \text{ contains } e_i \\ f^{c,i} & \text{otherwise} \end{cases}, \ \forall e_i \in E$$

$$F^p = \{f^{p,i} | i = 1 \ldots M\}$$

Table 4.1: *Network status notation*

Notation	Meaning
$f^{c,i}$	The current number of wavelengths used in link e_i before provisioning
F^c	Set of $f^{c,i}$
μ^c	Mean of F^c
$(\sigma^c)^2$	Variance of F^c
$p_w^{c,j}$	The total of current number of wavelength channels on links contained in path p_j
$f^{p,i}$	The number of wavelengths used in link e_i after provisioning
F^p	Set of $f^{p,i}$
μ^p	Mean of F^p
$(\sigma^p)^2$	Variance of F^p
$p_w^{p,j}$	The total number of wavelength channels on links contained in path p_j after provisioning

- *Mean*

  Mean of the current congestion level of F^c is:

$$\mu^c = \frac{1}{M}\sum_{i=1}^{M} f^{c,i}$$

Assume route p_j traverses through H links $(e_j^1, \ldots, e_j^H)$, then the total number of wavelength channels used in these link is:

$$p_w^{p,j} = \sum_{k=1}^{H} f_j^{p,k} = \sum_{k=1}^{H}(f_j^{c,k} + 1)$$

or

$$p_w^{p,j} = \sum_{k=1}^{H} f_j^{c,k} + H$$

where $f_j^{c,k}$ and $f_j^{p,k}$ are the congestion of link e_j^k before and after provisioning, respectively.

The mean of link congestion index in the network, thus, after provisioning is:

$$\begin{aligned}\mu^p &= \frac{1}{M}\sum_{i=1}^{M} f^{p,i} \\ &= \frac{1}{M}\sum_{i=1}^{M} f^{c,i} + \frac{H}{M}\end{aligned}$$

Thus,

$$\mu^p = \mu^c + \frac{H}{M} \tag{4.11}$$

Equation 4.11 relates the average congestion in the network before and after a connection is established. It is easy to conceive that network congestion level is higher and higher for each connection provisioned. If a provisioned route has H hops then mean of congestion rises by $\frac{H}{M}$. Since $0 < H \leq M$, hence, $0 < \frac{H}{M} \leq 1$. In our work, there are K routes for each connection and these routes have equivalent costs. Therefore, from the above observation, whichever of these routes is selected, *the increment of the mean is a constant and hence we cannot use this quantity to identify the better route.* This is, however, still a key quantity to calculate and investigate other factors.

- *The increment of f_{max}*

  The first priority for provisioning a connection is to maintain the current network congestion level. In fact, the increment value of f_{max} is either 0 or 1.

  Let p_1 and p_2 be two candidate routes for connection t. Assume that the cost of these routes are the same and equal to H. If a route (assume p_1) contains a link whose congestion index is equal to the congestion level and another route p_2 is not, then p_1 is selected. In the case that both p_1 and p_2 contain links whose congestion is network congestion level or do not contain such links at all, which route is said to be better in the sense of network congestion level? the question cannot be answered by this factor.

- *Standard deviation*

  Standard deviation σ shows the variance of link congestion index around the average congestion μ. In this sense, the maximal distance between links congestion and mean μ is proportional to σ. In other words, the number of links whose congestion index are equal to the congestion level is higher and higher with the increase of standard deviation. This trend leads to the fact that the probability of the increment of network congestion level is higher in next connections. Our objective is to maintain the standard deviation to be as low as possible.

Let p_j be a candidate route for a connection t, then these routes can be represented as follows:

$$p_j = (e^j_1, e^j_2, \dots, e^j_H)$$

where e^i_k is the link e_k contained in path p_j and H is cost of p_j.

The standard deviation $(\sigma^p_j)^2$ of F^p_j after path p_j is added is represented through the formula:

$$(\sigma^p_j)^2 = \frac{1}{M}\sum_{i=1}^{M}(f^{p,i}_j - \mu^p_j)^2$$

where $f^{p,i}_j$ is congestion of link e_i after adding path p_j.

Therefore, the difference of variances resulting from paths p_i and p_j is:

$$(\sigma^p_i)^2 - (\sigma^p_j)^2 = \frac{1}{M}\left(\sum_{k=1}^{M}(f^{p,k}_i - \mu^p_i)^2 - \sum_{k=1}^{M}(f^{p,k}_j - \mu^p_j)^2\right)$$

or,

$$(\sigma^p_i)^2 - (\sigma^p_j)^2 = \frac{2}{M}\sum_{k=1}^{H}(f^{c,k}_i - f^{c,k}_j) \tag{4.12}$$

Equation 4.12 shows that the difference between variances before and after a connection is provisioned only dependent on the total number of wavelength channels in links contained on that paths. It can be concluded that the increment of standard deviation affected by adding a new route is in a direct ratio to the total number of wavelength channels used in links along the path. *This is also a key in our work to justify a better path with respect to congestion level, i.e the total number of wavelength channels p^c_w used on a route before the route is provisioned.* Amongst candidate routes, the route whose p^c_w is minimum will be selected.

In short, given candidate routes $(p_1, p_2, \dots, p_K)$ for provisioning a connection, let $p^{c,i}_w$ be the total number of wavelength channels used on route p_i before the route provisioned. Route p_s is selected if $p^{c,s}_w = \min_{i=1\dots k}(p^{c,i}_w)$.

4.4 Approaches To Survivable Logical Topology Design

In this section, we investigate network survivability of the logical topology through two main approaches in the literature, namely ILP formulation and graph theory. We analyse the strengths and weaknesses of these approaches and propose a new heuristic approach that can benefit from advantages of the ILP model and the graph theory. Before more details of these approaches are discussed, we summarize more specific in two key issues in our problem, including optimization objective and capacity allocation for dedicated and shared protection schemes.

- Optimization objective

  As discussed in Section 4.3, our optimization objective function is an integration of total capacity used and congestion level and defined as:

  $$f_{com} = f_{sum} + \alpha f_{max}$$

  In this study, we mainly focus on minimizing the total capacity used in the network. Minimizing network congestion level is used to choose the lowest congestion solution among possible minimal feasible solutions. The weighting factor α, therefore, must be chosen to be less than $\frac{1}{W}$. In our work, we choose $\alpha = \frac{1}{W+1}$, and hence the objective function f_{com} is given by:

  $$f_{com} = f_{sum} + \frac{1}{W+1} f_{max}$$

  or,

  $$f_{com} = \frac{1}{W+1}\left((W+1)f_{sum} + f_{max}\right)$$

  Let f be defined as:

  $$f = (W+1)f_{sum} + f_{max} \tag{4.13}$$

  Then,

  $$f_{com} = \frac{1}{W+1} f$$

  Hence, if f is minimum, then f_{com} is minimum.

- Capacity allocation for dedicated and shared protection schemes

  The provisioning of traffic requirements is from connection to connection. However, the increment of network congestion level after each connection is provisioned differs between dedicated protection scheme and shared protection scheme. Link congestion is recalculated after a connection is provisioned in the following manner:

  1. *Dedicated scheme:*

     If a link is traversed on a working route or a backup route, then the number of available wavelength channels used in this link is increased by one. If the number of the available wavelength channels is zero then this link is busy or blocked, and hence no more connections can be routed over this link.

2. *Shared scheme:*

   The allocation wavelength channels for working paths is similar to dedicated scheme. However, the principle of capacity allocation for backup path is different. When the link is traversed on a backup path, the number of wavelength channels used in this link is increased by 1 if the proportion of the allocated back up channels over the total backup channels required in this link does not satisfy the sharing condition. Otherwise, the required wavelength channel is shared with other backup channels allocated on this link.

In short, let w_i and r_i be the total capacity required of working routes and backup routes on link e_i respectively; and f_{sum} and f_{max} be the total capacity allocated for these routes and network congestion level. Then f_{sum} and f_{max} will differ between dedicated and shared protection schemes as defined below.

1. *For dedicated protection schemes:*

$$f_{sum} = \sum_{e_i \in E} w_i + \sum_{e_i \in E} r_i \quad \text{and } f_{max} = \max_{e_i \in E}(w_i + r_i) \tag{4.14}$$

2. *For shared protection schemes*

$$f_{sum} = \sum_{e_i \in E} w_i + \sum_{e_i \in E} \lceil \frac{P}{Q} r_i \rceil \quad \text{and } f_{max} = \max_{e_i \in E}(w_i + \lceil \frac{P}{Q} r_i \rceil) \tag{4.15}$$

4.4.1 Integer Linear Programming (ILP) Formulation

Integer Linear programming is an useful tool to solve the SLTD problem. The advantage of this approach is that the establishment of all required connections is jointed and solved simultaneously. In addition, the primary and backup paths are also routed at the same time. Thus, ILP model can offer the expected optimal solutions. In this section, we propose some ILP models for path and link protection. The difference of our ILP model from those in the literature will be discussed.

As known, survivable routing in our ILP model has to discover primary paths and backup paths so that the restorability of the network against single link failures is one hundred percent. The routes, of course, have to satisfy conditions such as network constraints, flow constraints and protection constraints. In addition, the routes must be established so that the maximal congestion (blocking) on fiber links is minimum. More details, when the problem is modeled as Integer Linear Programming (ILP) formulation, is given here.

Path protection

We first discuss the ILP model for the path protection scheme. For each connection, we have to find two link-disjoint paths (routes), i.e working path and backup path. A path is modeled through the link indicate variables. For example, w^i_{+j} and w^i_{-j} are variables used to model link e_j contained on the working path of connection t_i. These are zero-one variables and denote that w^i_{+j} is equal to 1 if connection t_j traverses through link e_j in forward direction, otherwise w^i_j is equal to 0. Therefore, the total number variables in the ILP model is $2MD + 1$. In path protection, the allocation of network capacity (or wavelength channels) is different between dedicated and shared schemes and hence the capacity constraint in our ILP model also differs between these two schemes.

Objective:

$$\text{Minimize: } f = (W + 1)f_{sum} + f_{max} \tag{4.16}$$

where f_{sum} and f_{max} differ between dedicated protection scheme and shared protection scheme as in Equation 4.14 and Equation 4.15.

Subject to:

- *Flow conservation constraint:*

  For each connection t_i,

$$\sum_{e_j \in IE_k} (w^i_{+j} - w^i_{-j}) = \begin{cases} 1 & \text{if } v_k \text{ is the source node} \\ -1 & \text{if } v_k \text{ is the destination node} \\ 0 & \text{otherwise} \end{cases}, \quad \forall k \in V \tag{4.17}$$

$$\sum_{e_j \in IE_k} (r^i_{+j} - r^i_{-j}) = \begin{cases} 1 & \text{if } v_k \text{ is the source node} \\ -1 & \text{if } v_k \text{ is the destination node} \\ 0 & \text{otherwise} \end{cases}, \quad \forall k \in V \tag{4.18}$$

  The number of constraints is $2ND$

- *Congestion constraint:*

  The constraint differs between dedicated and shared schemes:

  - Dedicated schemes:

$$\sum_{t_i \in T} (w^i_{\pm j} + r^i_{\pm j}) \leq f_{max}, \forall e_j \in E \tag{4.19}$$

    The number of constraints is M

- Shared schemes

$$\sum_{t_i \in T} w^i_{\pm j} + \lceil(\frac{P}{Q}\sum_{t_i \in T} r^i_{\pm j})\rceil \leq f_{max}, \forall e_j \in E \tag{4.20}$$

where Q backup paths can share P wavelength channels ($\frac{P}{Q}$) and D is the total traffic connections required.

The number of constraints is $2M$

- *Capacity constraint:*

$$f_{max} \leq W \tag{4.21}$$

The number of constraints is 1

- *Path protection constraint:*

$$w^i_{+j} + w^i_{-j} + r^i_j + r^i_{-j} \leq 1, \quad \forall t_i \in T,\ e_j \in E \tag{4.22}$$

The number of constraints: M

- *Integer constraint:*

$$w^i_{\pm j} = \{0,1\}, \quad \forall t_i \in T, e_j \in E \tag{4.23}$$

$$r^i_{\pm j} = \{0,1\}, \quad \forall t_i \in T, e_j \in E \tag{4.24}$$

$$f_{max} \geq 0, \text{ integer} \tag{4.25}$$

Equation 4.16 states the objective function f_{com} of our ILP formulation. This is an integrated objective of capacity used and network congestion level. f_{com} is an integer number larger than or equal to 0; it is not a zero-one variable. Equation 4.17 and Equation 4.18 assure the conservation of working and backup flows in the network. The congestion metric is measured as maximal congestion of network links and modeled through Equation 4.19 for dedicated protection schemes, and Equation 4.20 for shared protection schemes. For dedicated schemes, congestion of a link is the total number of wavelength channels used in both primary and backup routes. For shared schemes, the congestion is the total number of wavelength channels used for primary routes and a partition of wavelength channels used for backup routes. The partial number is calculated according to the sharing ratio $\frac{P}{Q}, P < Q$, which denotes that Q wavelength channels for backup routes can share P physical wavelength channels. Since the congestion metric denotes the maximum number of wavelength channels of network links, the capacity constraint only requires the congestion variable f_{max} to be no larger than the available wavelength channels in the network as in Equation 4.21. Note that in our study we assume the available wavelength channels on all networks link are the same. Path protection constraint is represented in Equation 4.22 which assures that the working and backup paths of a connection are link-disjoint. Finally, Equations 4.23, 4.24 and 4.25 are the integer constraints for indicated variables; link indicated variables w_j^i and r_j^i are zero-one variables whereas the congestion variable f_{max} is non-negative integer.

4.4.2 Link protection

Since our objective and assumptions for link protection and path protection schemes are the same, the ILP model for link protection similar to those for path protection. However, because the operational mechanism of link protection differs from those of path protection, the ILP model need to be modified. The number of link variables for working paths is not changed but link variables for backup paths need to be re-defined for each working channel. Hence the number of variables in link protection schemes is larger than in path protection. For each connection, the number of variables for working routes is equal to M and for each working channel, the number of variables for backup path is M. Therefore, the total number of variables for each connection is M^2 and the total number of variables for all D traffic connections is DM^2.

The objective of ILP model for link protection is also to minimize a combined objective of total capacity used, f_{sum}, and the maximum congestion f_{max} in the network. Since backup routes are defined for specific working channels, these backup paths themselves are disjoint with their primary working channels. Therefore, the flow conservation and protection constraints can be combined into one constraint. The rest of the constraints are similar to the ILP model for path protection. The ILP model is mathematically represented as below.

Objective:

$$\text{Minimize: } f_{com} = (W+1)f_{sum} + f_{max} \tag{4.26}$$

Obviously, f_{sum} and f_{max} are different between dedicated protection scheme and shared protection scheme as in Equation 4.14 and Equation 4.15.

Subject to:

- *Flow conservation and protection constraints:*

  For each connection t^i, the flow conservation constraint is different.

- *Working paths*

$$\sum_{e_j \in IE_k} (w^i_{+j} - w^i_{-j}) = \begin{cases} 1 & \text{if } v_k \text{ is a source node} \\ -1 & \text{if } k \text{ is a destination node} \\ 0 & \text{otherwise} \end{cases}, \quad \forall k \in V \qquad (4.27)$$

- *Backup paths*

  The flow conservation of each required wavelength channels of working paths is modeled separately between forward direction and backward direction.

$$\sum_{e_l \in IE_k - \{e_j\}} (r^i_{+j+l} - r^i_{+j-l}) = \begin{cases} 1 & \text{if } v_k \text{ is the start node of } e_j \\ -1 & \text{if } v_k \text{ is the end node of } e_j \\ 0 & \text{otherwise} \end{cases}, \quad \forall t_i \in T, \forall k \in V \qquad (4.28)$$

$$\sum_{e_l \in IE_k - \{e_j\}} (r^i_{-j+l} - r^i_{-j-l}) = \begin{cases} 1 & \text{if } v_k \text{ is the end node of } e_j \\ -1 & \text{if } v_k \text{ is the start node of } e_j \\ 0 & \text{otherwise} \end{cases}, \quad \forall t_i \in T, \forall k \in V \qquad (4.29)$$

The number of constraints is $ND + MDN$

- *Congestion constraint:*

The constraint differs between dedicated and shared schemes:

- Dedicated scheme:

$$\sum_{t_i \in T} w^i_{\pm j} + \sum_{e_k \in E - \{e_j\}} \sum_{t_i \in T} r^i_{\pm j \pm k} \leq f_{max}, \forall e_j \in E \qquad (4.30)$$

  The number of constraints is M

- Shared scheme:

$$\sum_{t_i \in T} w^i_{\pm j} + \sum_{e_k \in E - \{e_j\}} \lceil \sum_{t_i \in T} \frac{P}{Q} r^i_{\pm j \pm k} \rceil \leq f_{max}, \forall e_j \in E \qquad (4.31)$$

  where Q backup paths can share P wavelength channels ($\frac{P}{Q}$) and D is the total traffic connections required.
  The number of constraints: $2M$

- *Capacity constraint:*

$$f_{max} \leq W \qquad (4.32)$$

The number of constraints: 1

- *Integer constraint:*

$$w^i_{\pm j} = \{0,1\}, \quad \forall t_i \in T,\ e_j \in E \tag{4.33}$$

$$r^i_{\pm j \pm k} = \{0,1\}, \quad \forall t_i \in T,\ e_j \in E,\ e_k \in E \tag{4.34}$$

$$f_{max} \geq 0, \text{ integer} \tag{4.35}$$

The objective function that minimizes the combined objectives of the total capacity used and the maximum congestion is represented in Equation 4.26. Equation 4.27 states the flow conservation constraint of working routes whereas the flow conservation of backup paths and protection constraints are combined in Equations 4.28 and 4.29. Similar to path protection, the congestion constraint differs between dedicated and shared schemes as in Equation 4.30 and 4.31. The capacity and integer constraints are denoted in Equations 4.32, 4.33, 4.34 and 4.35

The Post-ILP stage

All required connections have been allocated on network links in ILP formulation stage. In other words, the result of ILP formulation contains binary link indicator variables of working routes and backup routes of connections. In this part, we devise a simple algorithm to point out the exact route of connections that can be represented as a sequence of network nodes $[v_1 = v_s, v_2, \ldots, v_{k-1}, v_k = v_d]$.

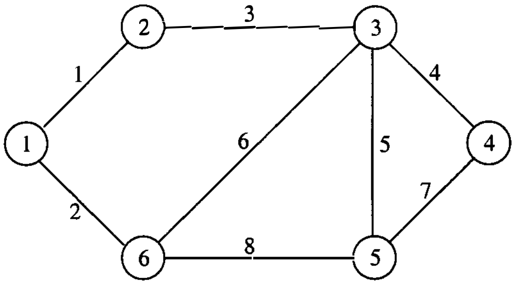


Figure 4.3: *A physical topology*

Table 4.2: *Route conversion table*

Required	ILP Results								Route
Connections	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	Conversion
(1,4)	1	0	1	1	0	0	0	0	[1 2 3 4]
(2,5)	0	0	1	0	1	0	0	0	[2 3 5]
(3,6)	0	0	0	0	0	1	0	0	[3 6]

Table 4.2 illustrates the results of path conversion through the network given by Figure 4.3. The first column indicates three required connections of traffic requirements. The second column shows the result of the ILP stage. This result is presented by link indicator variables. The Post-ILP stage converts the representation of ILP result to explicit form in column 3. For example, the route of first connection traverses through links $\{1, 3, 4\}$ from source node 3 to destination node 4 as a result of the ILP formulation stage. This route can be represented as a sequence of nodes $[1 - 2 - 3 - 4]$ in the post-ILP stage through the algorithm below, called *Path Conversion.* Note that the algorithm is applicable for single routes. In order to convert all D routes of traffic requirements, we apply the algorithm D times.

Algorithm 6 *Path Conversion*

```
Input : A ILP route of the connection established by the ILP stage
Output: a converted route of the ILP route
  Step 1: - $E_p$ is set of links in the ILP route;
          - Initialize route $P \leftarrow$ source nodes $s$;
          -  next node $\leftarrow s$
  Step 2: - Find a link $e$ in $E_p$ that contains the next node;
  Step 3: - Add the other end-nodes $v_k$ of link $e$ to $P$;
          - Assign next node $\leftarrow v_k$
  Step 4: - Repeat step 2 and 3 until next node $= d$
```

4.4.3 Graph Theory Approach

ILP formulation offers a mathematical model to obtain accurate solutions to the SLTD problem. Unfortunately, this has been proven to be NP-hard and intractable even with moderate scale networks. Heuristic approaches based on algorithms of graph theory have been studied intensively in the literature [12, 15]. Most of the approaches in graph theory are based on two common approaches, namely the *two-step approach* and the *one-step approach.* In this section, we investigate these approaches and develop algorithms that commit with our assumptions of network environment, traffic and protection requirements. Furthermore, our objective is to minimize the number of wavelength used and network congestion level.

Two-step approach

Under protection context, the process of finding working paths and backup paths can be separately performed as a two steps process, called *two-step approach.* The provisioning of traffic requirements in this approach is a sequence of connection-by-connection. For provisioning of each connection, the basic idea of the approach can be summarized as follows:

- **Step 1:** Find working path over the given physical graph from the source node to the destination node. The route is determined according to objectives. For instance, if the objective is to minimize the number of hops then a shortest path between end-nodes of the connection should be chosen. On the other hand, the working path may be routed to satisfy the objective of minimizing network congestion level.
- **Step 2:** All network links traversed by the working path are removed out of the network and the backup path is routed in the reduced graph from source to destination nodes. Again, the route is determined in the same way as the first step.

The two-step approach can be applied to two scenarios of network protections, namely path protection and link protection. In each scenario, the approach is modified to adapt to the different schemes of allocation wavelength channels for backup paths, namely dedicated scheme and shared scheme.

One-step Approach

Two-step approach is simple both in theory and application. In practice, this often results in a min-cost disjoint path-pair, but in general, it is not guaranteed that total cost of the path-pair is *minimum* and *even feasible*, as shown in Chapter 2. An alternative solution that resolves these two issues is proposed by Bhandari [25], known as the *one-step approach.* This approach is not much more complex than the two-step approach but is guaranteed to find the min-cost of pair-path [3]. This approach is referred on Surablle's algorithm. This algorithm, however, only concerns the minimum cost of disjoint path-pair problem with positive weight of links in the network. Bhandari's algorithm is much simpler than Surablle's algorithm and can handle negative weight of links through his modification to Dijkstra's algorithm. This approach is adopted from [3] and summarized as follows:

1. Take a shortest path between the source, s and target d. Denote this as p_1.

2. Define the direction of each edge traversed in p_1 from s toward d as positive.

3. Remove all directed edges on the shortest path p_1 and replace them with reverse direction edges by multiplying the original costs by -1.

4. Find the least cost path from s to d in the modified graph using the modified Dijkstra algorithm. Denote this path p_2.

5. Remove any edge of the original graph traversed by both p_1 and p_2. These are called *interlacing* edges. Identify all path segments identified by the edge removal from path p_1 and p_2.

We notice that the difference between the one-step approach from the two-step approach is that in the one-step approach, a working path and its backup are routed simultaneously. In addition, this approach is more applicable for path protection than link protection. In fact, in case of link protection and for the same conditions of the network, the one-step and the two-step approaches result in the same solution. Therefore, path protection will be investigated in both two-step and one-step approaches. Meanwhile, link protection is only investigated with the two-step approach.

Path protection

Path protection requires finding two disjoint paths between source and destination nodes of traffic connections. The two-step and the one-step approaches applied to this task are as follows: 1) The numerous of candidate minimal routes are discovered over the network; 2) An unique route must be selected among these candidate routes so that the congestion level in the network is affected the least. The principle to select the unique route follows Equation 4.12. More details on these approaches to path protection is as follows:

1. *Two-step approach*

   In this approach, working routes and backup routes are provisioned separately. For each connection, a working route is first provisioned and then a backup path is provisioned. The provisioning process of working paths and backup paths, however, is not different. Therefore, we first propose an algorithm, called *route provisioning algorithm.* This algorithm is used to provision both working routes and backup routes. Next, a complete algorithm of the two-step approach is developed to provision all required traffic connections.

   (a) *Route provisioning algorithm*

Route provisioning algorithm includes two-step. In the first step, a set of K minimum candidate routes between a source node and a destination node is generated. Next, a route that is the most suitable is selected as the provisioned route.

K shortest paths problem has been intensively studied in the literature [37, 34, 35, 36]. These have usually been implemented with a given constant K. As a result, the first K shortest paths found are yielded. Our problem, however, is different. We attempt to find the first K shortest paths whose costs are equivalent to each other. In other words, the value of K is varying from connection to connection. We, therefore, adopt and modify the K shortest paths algorithm in [35] to serve our purpose. A counting variable is introduced to monitor the number of shortest paths found. The algorithm is terminated when the cost of a next shortest path is larger than those of its previous paths. The pseudo-code of the algorithm is presented as in Algorithm 7

Therefore, K candidate minimum routes for a connection from s to d is obtained from Algorithm 7. The provisioning process is complete by using the next algorithm presented in Algorithm 8. This algorithm shows how to select a route among the K candidate routes.

(b) *Path protection algorithm: the two-step approach*

Route provisioning procedure is operated as the core of the two-step algorithm. In this procedure, connections are provisioned sequentially and, for each connection, the working route is provisioned and followed by provisioning the backup route. A pseudo-code of the two-step approach for the path protection is represented in Algorithm 9

2. *One-step approach*

The one-step approach, basically, operates as same as the two-step approach, except that a working route and a backup route of a connection are discovered simultaneously instead of separately. This difference allows us to find K minimum pairs of disjoint paths, so-called K *minimum disjoint path-pairs*, instead of K shortest path as in two-step approach. To our best knowledge, there is only Bhandari's algorithm to find a minimum disjoint paths. This algorithm is not applicable for finding K minimum disjoint path-pairs. In this body of work, we base on Bhandari's algorithm to devise a new algorithm for finding K minimum disjoint path-pairs. This algorithm is named as K*-disjoint path-pairs* and summarized as follows:

Algorithm 7 *K shortest paths*

```
Input : - Physical topology $G(V, E)$;
        - $(s, d)$ source and destination nodes, respectively, of a traffic connection
Output: - $K$, the number of shortest paths found;
        - Route table of the $K$ shortest paths found.
1: To assure the possible repetition in a path of the initial and destination nodes,
   the given network is enlarged with a super initial node $S$, and super destination
   nodes $T$, with zero cost arcs $(S, s)$ and $(t, T)$. We find the shortest tree from source
   node $s$ to other nodes in the network and mark the shortest path $p_1 = \{s_0(= s), s_1, \dots, s_{r-1}, s_r(= d)\}$
   from $s$ to $d$ as the first shortest path.
2: Determine the first node $s_h$ in $p_1$ such that $s_h$ has more than a single incoming arc.
   If there does not exist $s'_h$, then the node $s'_h$ is generated, else determine next node
   $s_i$ in $p$ that has not alternate yet. The incoming arcs of node $s'_h$ are incoming arcs
   of $s_h$, except those coming from $s_{h-1}$. The shortest path from $s$ to $s'_h$ $(d(s, s'_h))$ is
   calculated as:
        $d(s, s'_h) = \min_x\{d(s, x) + d(x, s'_h)\}$,
   where $(x, s'_h)$ are incoming arcs of $s'_h$.
3: For each $s_j \in \{s_i, \dots, s_{r-1}\}$, generate $s'_j$ following the same rules as $s'_h$, but with
   one more incoming arc of $(s'_{j-1}, s'_j)$. Clearly, the shortest path from $s$ to $s'_j$ is the
   second shortest path from $s$ to $s_j$. Therefore $p_2 = \{s_0, \dots, s'_i, \dots, s'_{r-1}, s_r(= d)\}$ is
   the second shortest path.
4: If the cost of $p_2$ is larger than the cost of $p_1$, then the algorithm is terminated,
   otherwise go to step 2 for shortest path $p_k(k = 2, 3, \dots)$ to find the next shortest
   path until the cost of $p_k$ is larger than the cost of $p_{k-1}$.
```

Algorithm 8 *Route selection for minimum congestion*

```
Input : - Working and backup route tables: $WRT$ and $BRT$ and network congestion
    level $f_{max}$;
        - $P$ set of $K$ candidate routes;
Output: - Route $p^s$ is selected from $P$;
    Find the total capacity used $p_w^{p,i}$ to all candidate routes in $P$ after these routes
    are assumed to be selected; the calculation of the total capacity used for dedicated
    protection and shared protection schemes follow Equation 4.14 and Equation 4.15,
    respectively.
    Select routes $p^j$ that have minimum $f_w^{s,i}$. This will first select routes which do not
    traverse through links whose congestion is less than current network congestion
    level. If all routes traverse through such links whose congestion is equal to current
    network congestion level or less than current network congestion level at all, then
    all of these routes are selected in this step
    Select a route $p_s$ so that $p_w^{c,s}$ is minimum among $p_w^{c,j}$. We notice that there may be
    more than one such $p_s$. In this case, $p_s$ is randomly selected.
```

Algorithm 9 *Two-step approach - Path Protection*

```
Input : - A physical topology $G(V, E)$;
        - Traffic requirements $T$
Output: - Route tables for working path ($WRT$), and backup path ($BRT$);
        - Network congestion $f_{max}$
    Initialize $WRT \leftarrow \emptyset$; $BRT \leftarrow \emptyset$;
    For each connection $t_i \in T$
    Find the working path:
    This path is found by using Algorithm 7 and Algorithm 8;
    Update working route table $WRT$;
    Remove all link $e_j \in p_w^i$ out of $G(V, E)$ into $G_r(V, E)$;
    Find the backup path:
    Similar to finding the working path;
    Update backup route table $BRT$;
    Update $G(V, E)$ with the found working and backup route;
    Provision next connections
```

Algorithm 10 *The one step approach - K-disjoint path-pairs*

```
Take a shortest paths between the source $s$ and target $d$. Denote this $p$.
Define the direction of each link traversed in $p$ from $s$ toward $d$ as positive.
Remove all directed links on the shortest path $p$ and replace them with reverse direction links by multiplying the original link cost with -1.
Find $K$ least cost paths from $s$ to $d$ in the modified graph using Algorithm 7. Denote them as the set of path $S = \{s_1, s_2, \dots, s_K\}$.
For each pair of paths $(p, s_i)$, remove any link of the original graph traversed by both $p$ and $s_i$. These are called interlacing links. Identify all path segments identified by the link removal from path $p$ and $s_i$. These such path-pairs form $K$ disjoint path-pairs $\{(w_1, r_1), (w_2, r_2), \dots, (w_K, r_K)\}$.
```

Algorithm 10 results in K minimum disjoint path-pairs. The total capacity used in these path-pairs, however, may not be equivalent. Our objective is to select all path-pairs that use the same number of capacity units and are minimum. The process of provisioning a connection is completed by selecting a path-pair among such path-pairs. The selection follows the same procedure as the second step of provisioning a route, except that a route is now replaced by a path-pair.

Link protection

In link protection, backup routes of a connection are determined after its working is provisioned, hence the one-step approach is not applicable in this scheme of protection. In this part, we aim to implement link protection through the two-step approach. For each connection, a working route is first provisioned. Next, for each channel in the working route, a backup route is provisioned.

It can be easily seen that provisioning of working routes can be performed through *route provisioning algorithm* in the two-step approach for path protection. In addition, the provisioning of backup paths of a connection can also be performed through this algorithm with two differences:

- The link containing the working channel needs to be removed out of the graph.
- The route provisioning algorithm is applied to the modified graph between two end-nodes of that link.

The complete algorithm for link protection is summarized as follows:

Algorithm 11 *Two-step approach: Link protection*

```
1: For connection $t_i$,
2: Find $K$ minimum disjoint path-pair using Algorithm 10
3: Select the best path-pair among the $K$ path-pairs. This can be done using Algorithm 8 with one change in the input parameter, ie. $K$ path-pairs are used instead of $K$ shortest paths.
4: Go to step 1 until all connection is provisioned.
```

4.4.4 Our Proposed Approach to SLTD Problem

As mentioned in Chapter 2, the Survivable Logical Topology Design problem has been proven to be a NP-hard. The solution to this problem can be either *optimal* or *near-optimal.* The optimal solution can be obtained from Integer Linear Programming (ILP) or exhausting searching. These approaches are simple but intractable with increasing of the network size (network nodes and links). Although several heuristic approaches based on graph theory have been proposed to obtain near-optimal solutions with fast computation, it is still a challenge to achieve optimal solutions which are computational efficient.

Motivate by this challenge, we propose a heuristic approach for path protection that combines the computational advantages of graph algorithms and optimal solutions of ILP solver with small number of integer variables and constraints. The approach includes two steps: K minimum disjoint path-pairs (KDPPs) and ILP selection formulation (ILPS). In KDPP, for each connection, we generate the set of K minimum disjoint path-pairs as possible solutions for that connection. The purpose of the ILPS step is to select the suitable path-pairs in all sets of K candidate path-pair of traffic connections so that the integrated objective function of the total capacity used and congestion level, as in Equation 4.13, is minimum. The important constraint in this formulation is the *selection constraint.* This constraint ensures that only one suitable path-pair among K candidate path-pairs of each connection is selected.

Network Model and ILPS Notation

For clarity of our discussions, we notate our problem as follows.

- Let an undirected graph $G(V, E)$ be a physical network topology, where $V = \{v_1, v_2, \ldots, v_N\}$ is the set of N vertices representing network nodes, and $E = \{e_1, e_2, \ldots, e_M\}$ is set of M undirected edges representing the bi-directional optical fibers.

- Let W be the weight of each link e_l, representing the maximum number of wavelengths in the link.
- Let p be a path between two nodes in G, denoted by $p = [b_1, b_2, \ldots, b_M]$, where b_i is link indicator constant of link i, and is given by:

$$b_i = \begin{cases} 1 & \text{if path } p \text{ uses link i} \\ 0 & \text{otherwise} \end{cases}$$

- Let c be the cost of path p, determined by

$$c = \sum_{l=1}^{M} b_i$$

  In our model, the cost of a path is defined as the number of wavelength channels taken by that path.

- Let $T = \{t_1, t_2, \ldots, t_D\}$ be the set of D traffic requirements (traffic connections) over the network, where t_i denotes the connection between node pair (s_i, d_i).
- K denotes the number of candidate path-pairs between end-nodes of a connection.
- $P_i = \{p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(K)}\}$ is the set of K candidate primary paths of connection t_i, where $p_i^{(j)}$ denotes the j^{th} primary path of connection t_i.
- $R_i = \{r_i^{(1)}, r_i^{(2)}, \ldots, r_i^{(K)}\}$ is the set of K candidate backup paths of connection t_i, where $r_i^{(j)}$ denotes the j^{th} backup path of connection t_i.

KDPP Formulation

The approach to find K minimum disjoint path-pairs between two nodes in the network was proposed in Algorithm 10. In this part, we define the representation for result of this algorithm. These representation is used in the next step of ILPS.

Given a physical topology $G(V, E)$ of a network, a set of connections T and a constant $K \in \mathbb{Z}^+$, the main objective of KDPPs is to compute D sets of K minimum disjoint path-pairs (disjoint lightpath's path-pairs) corresponding to D traffic connections. These are represented by two constant matrices that contains all primary paths and backup paths of all connections, and, two path cost matrices that store the corresponding costs.

1. *The constant matrices*

Let P and R represent the constant matrices for primary paths and backup paths respectively, then these are matrices of KD rows and M columns. We represent the constant matrices as follows:

- *The primary constant matrix*

$$P = \begin{pmatrix} P_1 & P_2 & \dots & P_D \end{pmatrix}^T, \qquad T : \text{Transpose}$$

where $P_i = \begin{pmatrix} p_i^{(1)} & p_i^{(2)} & \dots & p_i^{(K)} \end{pmatrix}^T, \forall i \in [1..D]$ are sub-matrices of $(K \times M)$ in which row $p_i^{(j)}$ represents the j^{th} path of connection t_i, and is expressed as:

$$p_i^{(j)} = [b_{p,1}^{(ij)}, b_{p,2}^{(ij)}, \dots, b_{p,M}^{(ij)}]$$

Note that in our study, path $p_i^{(j)}$ is modeled using link indicator constants $b_{p,l}^{(ij)}$, where $b_{p,l}^{(ij)}$ is defined as:

$$b_{p,l}^{(ij)} = \begin{cases} 1 & \text{if path } p_i^{(j)} \text{ uses link } l \\ 0 & \text{otherwise} \end{cases}$$

Therefore, sub-matrices P_i are represented as:

$$P_i = \begin{pmatrix} b_{p,1}^{(i1)} & b_{p,2}^{(i1)} & \dots & b_{p,M}^{(i1)} \\ b_{p,1}^{(i2)} & b_{p,2}^{(i2)} & \dots & b_{p,M}^{(i2)} \\ & \cdots\cdots\cdots & & \\ b_{p,1}^{(iK)} & b_{p,2}^{(iK)} & \dots & b_{p,M}^{(iK)} \end{pmatrix}, \forall i \in [1..D]$$

- *The backup constant matrix*

Similar to the primary constant matrix, the backup constant matrix is defined as:

$$R = \begin{pmatrix} R_1 & R_2 & \dots & R_D \end{pmatrix}^T, \qquad T : \text{Transpose}$$

where $R_i = \begin{pmatrix} r_i^{(1)} & r_i^{(2)} & \dots & r_i^{(K)} \end{pmatrix}^T, \forall i \in [1..D]$ are sub-matrices of $(K \times M)$ in which row $r_i^{(j)}$ represents the j^{th} backup path of connection t_i, and is expressed as:

$$r_i^{(j)} = [b_{r,1}^{(ij)}, b_{r,2}^{(ij)}, \dots, b_{r,M}^{(ij)}]$$

and path $r_i^{(j)}$ is modeled using link indicator constants $b_{r,l}^{(ij)}$, where $b_{r,l}^{(ij)}$ is defined as:

$$b_{r,l}^{(ij)} = \begin{cases} 1 & \text{if path } r_i^{(j)} \text{ uses link } l \\ 0 & \text{otherwise} \end{cases}$$

Hence, sub-matrices R_i are represented as:

$$R_i = \begin{pmatrix} b_{r,1}^{(i1)} & b_{r,2}^{(i1)} & \cdots & b_{r,M}^{(i1)} \\ b_{r,1}^{(i2)} & b_{r,2}^{(i2)} & \cdots & b_{r,M}^{(i2)} \\ & \cdots\cdots\cdots & & \\ b_{r,1}^{(iK)} & b_{r,2}^{(iK)} & \cdots & b_{r,M}^{(iK)} \end{pmatrix}, \forall i \in [1..D]$$

2. *The path cost matrices*

Path cost matrices denote the cost of primary and backup paths. The entries of these matrices are the number of wavelengths used in all possible lightpaths of primary and backup paths. These matrices are employed to model the objective function in ILPS.

Let C_P and C_R be the path cost matrices of candidate primary paths and backup paths respectively, these matrices are $(D \times K)$ matrices, and given by:

$$C_P = \begin{pmatrix} c_{1,1}^p & c_{1,2}^p & \cdots & c_{1,K}^p \\ c_{2,1}^p & c_{2,2}^p & \cdots & c_{2,K}^p \\ & \cdots\cdots\cdots & & \\ c_{D,1}^p & c_{D,2}^p & \cdots & c_{D,K}^p \end{pmatrix}$$

and,

$$C_R = \begin{pmatrix} c_{1,1}^r & c_{1,2}^r & \cdots & c_{1,K}^r \\ c_{2,1}^r & c_{2,2}^r & \cdots & c_{2,K}^r \\ & \cdots\cdots\cdots & & \\ c_{D,1}^r & c_{D,2}^r & \cdots & c_{D,K}^r \end{pmatrix}$$

where $c_{i,j}^p$ and $c_{i,j}^r$ denotes the number of wavelength channels taken by primary path $p_i^{(j)}$ and backup path $r_i^{(j)}$ respectively.

ILP Selection Formulation (ILPS)

The goal of ILPS formulation is to select suitable path-pairs from outcomes of the KDPPs step. The selection process has to satisfy the following conditions:

- The integrated objective function f in Equation 4.13 is minimized.
- For each connection, only one path-pair is selected.
- The total number of wavelength channels used per link does not exceed the link capacity.

Let $x_i^{(j)}$ be a path-pair indicator variable defined as:

$$x_i^{(j)} = \begin{cases} 1 & \text{if } t_i \text{ uses path-pair } (p_i^{(j)}\text{-}r_i^{(j)}) \\ 0 & \text{otherwise} \end{cases}$$

The objective of the ILPS is to minimize the total number of wavelengths used and the congestion level in the network, given by:

$$f = (W + 1) f_{sum} + f_{max}$$

where f_{sum} is a dependent variable and f_{max} is a decision variable. These differ between dedicated and shared protection schemes, and are given as:

- *Dedicated protection scheme*

$$f_{sum} = \sum_{i=1}^{D} \sum_{j=1}^{K} (c_{i,j}^{p} + c_{i,j}^{r}) x_i^{(j)}$$

$$f_{max} = \max_{e_l \in E} \left(\sum_{i=1}^{D} \sum_{j=1}^{K} (b_{p,l}^{ij} + b_{r,l}^{ij}) \right)$$

- *Shared protection scheme*

$$f_{sum} = \sum_{l=1}^{M} \left(\sum_{i=1}^{D} \sum_{j=1}^{K} b_{p,l}^{i,j} x_i^{(j)} + \lceil \frac{P}{Q} \sum_{i=1}^{D} \sum_{j=1}^{K} b_{r,l}^{i,j} x_i^{(j)} \rceil \right)$$

$$f_{max} = \max_{e_i \in E} \left(\sum_{j=1}^{D} \sum_{k=1}^{K} b_{p,i}^{jk} + \lceil \frac{P}{Q} \sum_{j=1}^{D} \sum_{k=1}^{K} b_{r,i}^{jk}) \rceil \right)$$

This formulation is subject to the following constraints:

1. Capacity constraint:

   This constraint is also different between dedicated and shared protection schemes, given as follows:

   - *Dedicated protection scheme*

$$\sum_{i=1}^{D} \sum_{j=1}^{K} (b_{p,l}^{(ij)} + b_{r,l}^{(ij)}) x_i^{(j)} \leq W, \qquad \forall l \in E$$

   - *Shared protection scheme*

$$\sum_{i=1}^{D} \sum_{j=1}^{K} b_{p,l}^{(ij)} x_i^{(j)} + \lceil \frac{P}{Q} \sum_{i=1}^{D} \sum_{j=1}^{K} b_{r,l}^{(ij)} x_i^{(j)} \rceil \leq W, \qquad \forall l \in E$$

2. Selection constraint:

$$\sum_{j=1}^{K} x_i^{(j)} = 1, \qquad \forall i \in [1..D]$$

3. Integer constraint:

$$x_i^{(j)} = \{0, 1\}, \qquad \forall i \in [1..D], j \in [1..K]$$

4.5 Numerical Results

In this section, the performance of our approach is compared with ILP formulation and graph theory approach using two performance metrics: the network performance and the implementation performance. The network performance, in this case, refers to network capacity utilization and network congestion level. The implementation performance, on the other hand, refers to computational time and the optimality of solutions.

4.5.1 The network performance

The total network capacity used f_{sum}, the network congestion level f_{max} and the integrated objective f_{com} are simulated and examined through three approaches: ILP formulation, graph theory and our proposed approach. In addition, capacity efficiency is also compared between dedicated path protection and dedicated link protection schemes.

Since our approach and the one-step approach in graph theory are only applicable for path protection, the comparison between f_{sum}, f_{max} and f_{com} is made for all approaches discussed above for path protection. For link protection, the comparison is made for two-step approach in graph theory. Table 4.3 summarizes the approaches that will be the subject to our comparison.

Network environment and simulation data are setup as follow:

Network environment and simulation data

Simulation in this part is performed over the typical topology NSFNET (the National Science Foundation Network) with $N = 14$ nodes and $M = 21$ links as in Figure 4.4. This is modeled as an undirected graph in which the capacity of each link is 16 wavelength channels.

Table 4.3: *The approaches is subject to simualtion*

	Path Protection			Link Protection		
	f_{sum}	f_{max}	f_{com}	f_{sum}	f_{max}	f_{com}
ILP formulation	✓	✓	✓	-	-	-
Two-step approach	✓	✓	✓	✓	✓	✓
One-step approach	✓	✓	✓	-	-	-
Our approach	✓	✓	✓	-	-	-

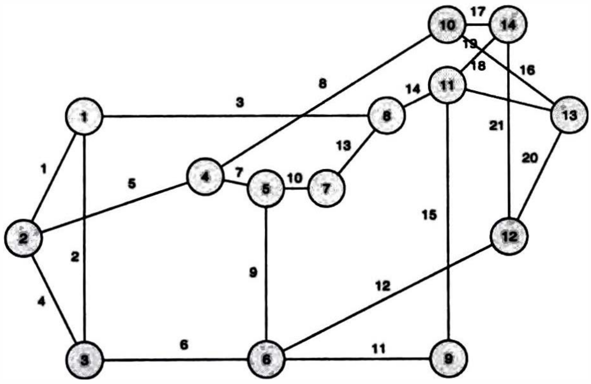


Figure 4.4: *NFSNET topology*

We randomly generate sets of traffic connections that are provisioned over this physical topology. We denote use NC the number of traffic connections. The number of traffic sets differs between different simulation. More details on traffic connection will be given for each simulation.

1. *The comparison of the network capacity usage (f_{sum}) and the congestion level (f_{max}) between three objectives*

   The purpose of this simulation is to compare and contrast the capacity efficiency (f_{sum}) and network congestion (f_{max}) for different optimization objectives, including minimizng the resource utilization, minimizng the congestion levels and minimizing the proposed integrated objective. For convenience, we name these objectives as *CapMin*, *CongMin*, and *CombMin*, respectively. In each case, we model path protection and determine the pair $\{f^u_{sum}, f^u_{max}\}$ for each objective $u \in \{CapMin, CongMin, CombMin\}$. We note that when the objective is, for example, to solely minimize the total used capacity (objective *CapMin*), we still need to measure both the total used capacity f^{CapMin}_{sum}, and the network congestion level f^{CapMin}_{max}.

   We simulate these three objectives through ILP formulation, two-step approach, one-step approach and our proposed approach. Our goal in this simulation is twofold. First, since ILP formulation offers exact solutions, we compare the performance of our objective function *CombMin* proposed in Section 4.3 over the existing objectives *CapMin* and *CongMin*. Secondly, with respect to the protection schemes, we wish to compare the network efficiency between path protection and link protetion over two-step and one-step approaches.

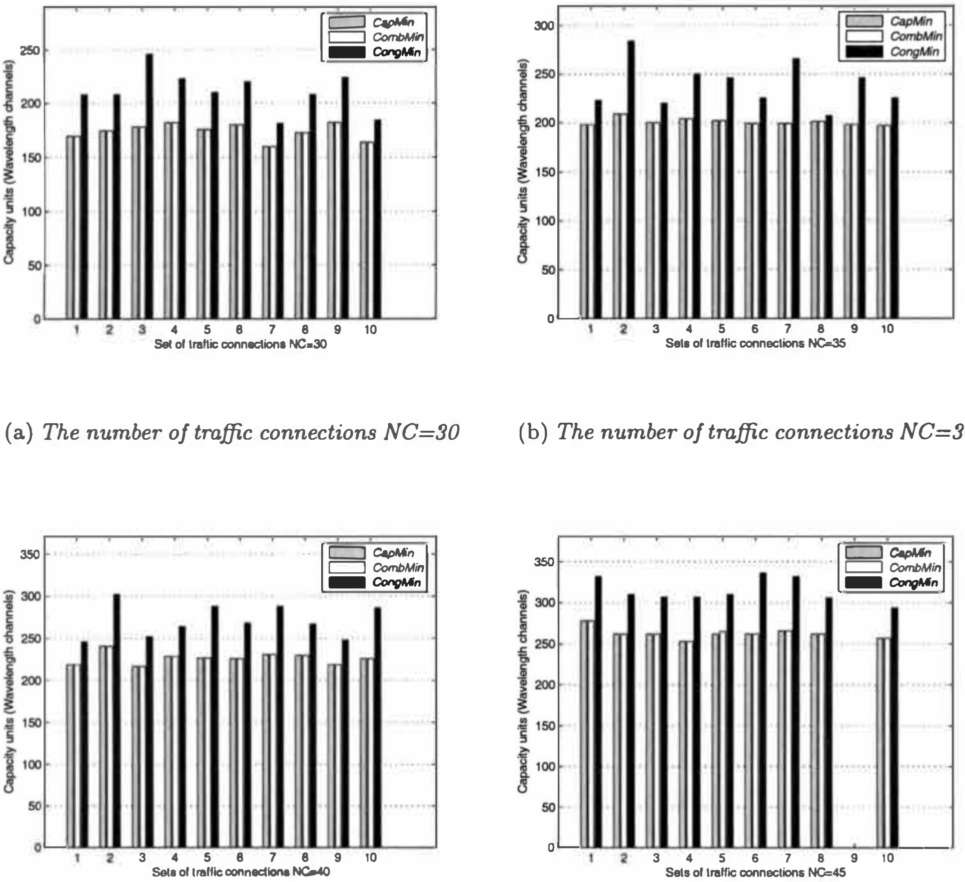
   The data set is generated as follows: 1) we use the typical physical topology NSFNET as described above; and 2) traffic connections are generated randomly with the number of traffic requirements is in the range of [30...45]. It is suspected that when the number of traffic connections is 30, optimum and feasible solutions are achieved through the approaches implemented since the number of capacity units required is low. However, when the number of traffic connections is higher (around 45), the provisioning for the random sets of traffic connections may succeed or fail due to the distribution of the generated traffic pattern.

   Analysis of the Results:

   The simulation results in which we compare the resource utilization (f_{sum}) and the congestion levels (f_{max}) are shown in Figure 4.5 and Figure 4.6.

   - The resource utilization:

     It can be seen in Figure 4.5 that the resource utilization corresponding to



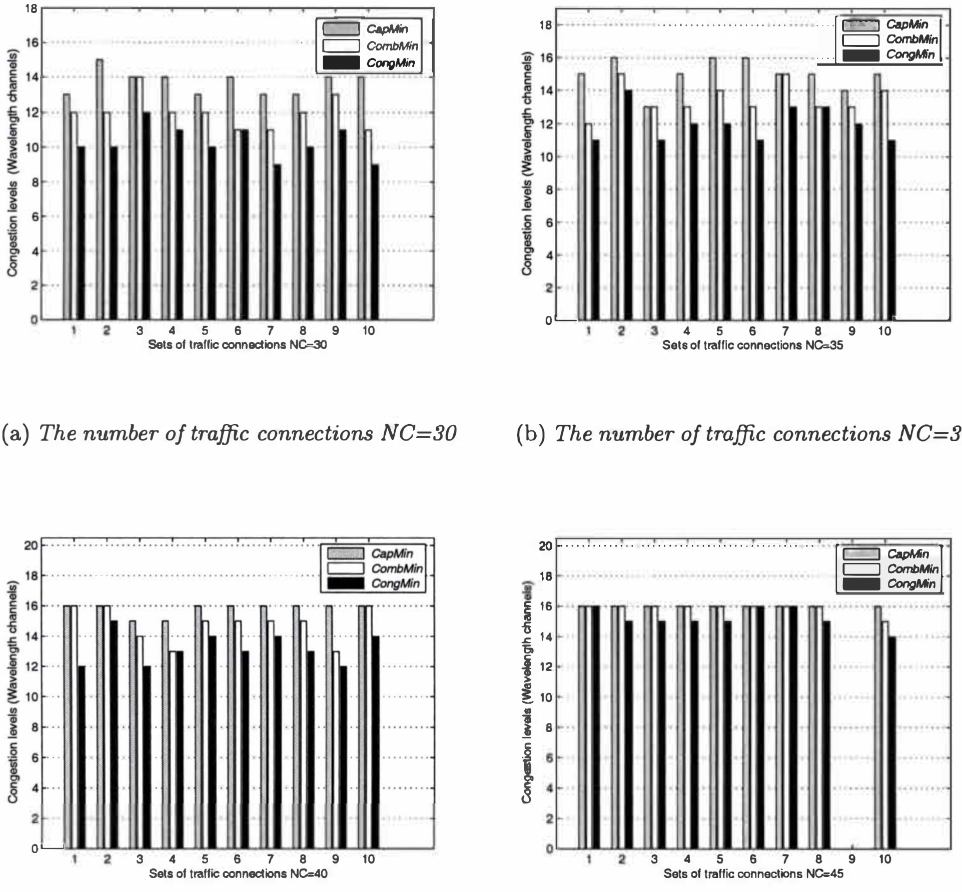
(a) *The number of traffic connections NC=30*

(b) *The number of traffic connections NC=35*

(c) *The number of traffic connections NC=40*

(d) *The number of traffic connections NC=45*

Figure 4.5: *Capacity utilization - Dedicated path protection - ILP formulation*



(a) *The number of traffic connections NC=30*

(b) *The number of traffic connections NC=35*

(c) *The number of traffic connections NC=40*

(d) *The number of traffic connections NC=45*

Figure 4.6: *Congestion levels - Dedicated path protection - ILP formulation*

  CombMin is always equal to those corresponding *CapMin*. These results validate the theory we developed in Section 4.3, that is, if the weighting factor α is less than $\frac{1}{W}$, then the resource utilization objective has a higher priority in optimization. In this study, we choose $\alpha = \frac{1}{W+H} < \frac{1}{W}$. Therefore, in terms of the resource utilization, the *CapMin* and *CombMin* always result in the same minimum capacity units used. The value of resource usage is also much better than *CongMin*. On average, the utilized resource for *CongMin* is required to be 1.2 times the utilized resource for the others objectives *CapMin* and *CombMin*. In other words, for the *CongMin* objective, results of the utilized capacity is around 120%, compared to the rest.

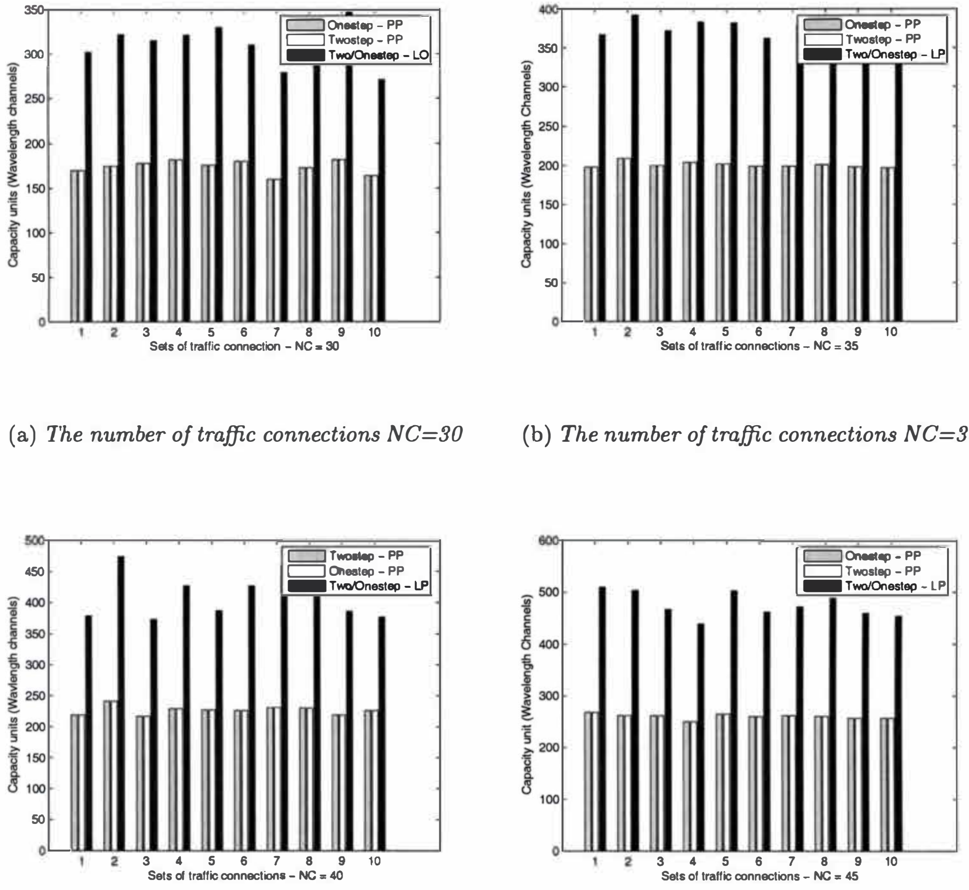
- The congestion levels:

  The best results for congestion levels in the network can be obtained through an ILP formulation in which the objective is to minimize congestion levels. This can be easy seen in Figure 4.6. However, as discussed, the required capacity is more than the other objectives *CapMin* and *CombMin*. We observe that the congestion levels achieved with *CombMin* objective is a little bit higher, compared to those with *CongMin*, but the values is always lower in case of the *CapMin* for all simulation with varying number of traffic connections. This, again, can be explained through the theory we proposed in Section 4.3. Therefore, the efficiency of congestion levels achieved with these objectives can be ranked from *CongMin*, *CombMin* and then *CapMin*.

2. *The comparison of capacity efficiency between link protection and path protection*

   The objective function proposed in this thesis has a priority of minimizing resource utilization and then the congestion levels. The two-step approach and the one step approach algorithms proposed attempt to achieve the objective through the theory for route provisioning developed in Section 4.3. These two algorithms implement the dedicated path protection and the dedicated link protection schemes. The simulation results are shown in Figure 4.7. The four subgraphs present the results when the number of traffic connections is 30, 35, 40, or 45. In each subgraph, there are three types of bars which present the number of utilized capacity units for path protection (PP) and link protection (LP) through two-step and one-step approaches. Note that since the operation mechanism of the two-step and the one-step approaches are the same for link protection, their results are shown in the same bar.

   The results have shown that resource utilization for path protection in two-step and one-step approaches are almost equivalent and also equivalent to the results



(a) *The number of traffic connections NC=30*

(b) *The number of traffic connections NC=35*

(c) *The number of traffic connections NC=40*

(d) *The number of traffic connections NC=45*

Figure 4.7: *Capacity requirements for path protection versus link protection*

we achieved with ILP formulation. The capacity required for link protection scheme are much higher than path protection. The number of capacity units used for link protection is about 182% more than the requirements of path protection scheme for the same network and traffic configurations.

4.5.2 The implementation performance

In this part, we examine our heuristic approach proposed in Section 4.4.4 in terms of two performance metrics: *the time complexity* and *the optimality of solutions*. Since the ILP formulation offers optimal solutions for survivable wavelength routing problem, we use these metrics as a comparison between our approach and the ILP formulation. *ComMin* is used as the objective function in both these examined approaches. The simulations are also carried out on NSFNET as described above. The ILP solver based on LP-relaxation and *branch and bound* techniques is developed in MATLAB environment.

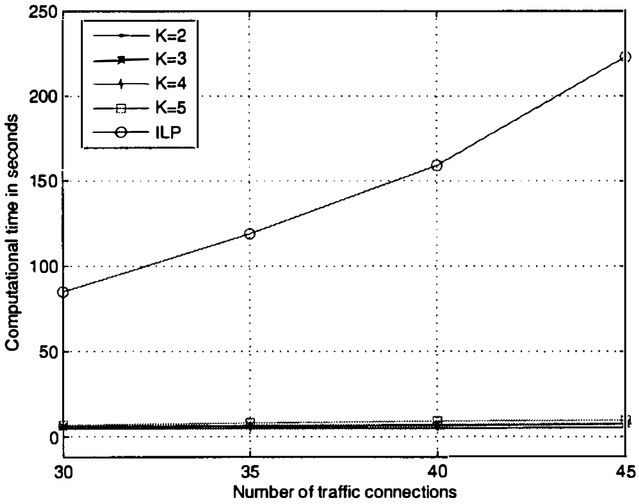


Figure 4.8: *Computational time versus the number of traffic connections*

1) *Time complexity:*

The computational time is measured in different scenario of traffic connections with a fix physical topology (NSFNET topology). The number of traffic connection D is varied from 30 to 45. Figure 4.8 shows the simulation results in comparison between our approach over the ILP formulation. For each value of D, we randomly generate 100 traffic matrices and the computational time is measured as the average computational time of these matrices. We observe an outstanding improvement in computational time of our approach compared to the classical ILP approach. The time complexity

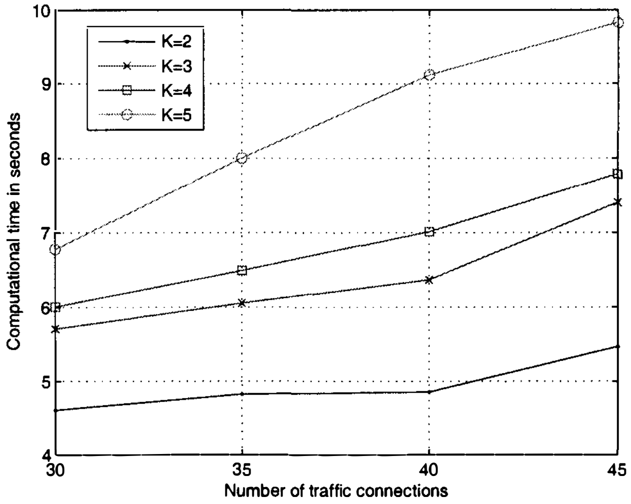


Figure 4.9: *Computational time versus the number of traffic connection - KDSP cases*

for the ILP rapidly increases (almost exponentially) while time complexity curves in our approach stay nearly flat. The computational time of ILP formulation at $D = 30$ connections is around 85 seconds and quickly increases to about 230 seconds when D=45 while the increase of those in our approach is not significant (in the range $[4 \ldots 9]$ seconds). In addition, it is worth noting the small differences for different values of K in our approach. The computational time only increases a couple of seconds for an increase of 1 unit in the value of K.

2) *Optimality of solutions:*

This simulation is also implemented in the undirected NSFNET in Figure 4.4. 50 traffic matrices are randomly generated for D from 30 to 45 connections. Our approach is implemented with $K = 1 \ldots 5$ and the outcomes are compared with the ILP formulation.

The results are presented in table 4.4 in which column 3 shows the number of feasible solutions achieved and column 4 represents the number of optimal solutions out of 50 randomly generated traffic requirements.

These results show that the number of feasible solutions and optimal solutions generally increase when K increases. With $K = 1$, the objective of our approach is basically to find the K shortest disjoint paths between node pairs (s, d) of traffic connections. This is only suitable for low traffic requirements. In fact, for $K = 1$, no feasible solutions are achieved in this simulation. With $K = 2$ our approach yields 10 of the 49 feasible solutions, and yields 44 out of the 49 feasible solutions when $K = 3$.

Table 4.4: *Comparing the number of optimal solutions between ILP and our approach*

	K	Feasible solutions	Optimal solutions
ILP	–	49	49
our proposed approach	1	0	–
	2	10	6
	3	44	39
	4	47	47
	5	49	49

In regard to the optimality of solutions, 100% of the solutions are optimal for $K \geq 4$, while for $K = 2$ and 3 the number of optimal solutions are 6 out of 10 and 39 out of 4 a respectively. In summary, the number of optimal solutions increases monotonically with K. This value can used to control the balance between the optimality of solutions and the computational time.

4.6 Concluding Remarks

We proposed an integrated objective function that can combine the two common objectives in the SLTD problem, namely minimizing the resource utilization and miniming the network congestion levels. These two objectives in the literature is treated separately. A *weighting factor* was introduced to control the balance between these objectives. For example, the network utilization can be given a higher priority by choosing a weighting factor less than $\frac{1}{W+1}$, where W is the maximum number of wavelength channels available. In our model, where we use the integrated objective, the results are significantly improved over other methods which only consider one objective. For instance, the network utilization in the combined objective case is equivalent to the outcome of those techniques which only aim at minimizing the total number of wavelength used, but the congestion level of our technique is much better. Furthermore, although the congestion level achieved in our combined objective case is a little bit higher than those algorithms which only attempt to minimize the congestion level, we achieve a much better network utilization as a result.

In addition, although SLTD has been extensively studied in the literature, it still remains a difficult problem and has been proved to be NP-complete. Time complexity and optimality of solutions are two conflicting metrics for assessing the outcome of SLTD optimization, and it is important to find approaches that can balance these. In this thesis, we proposed a two-step approach that combines the time complexity advantages of existing graph algorithms and optimal solution advantages of the ILP formulation. In the first step, a KDPP algorithm was proposed to find the K-disjoint path-pairs between a source node and a destination node of a connections. The second step was an ILP formulation, but the number of variables was significantly reduced, compared to the original ILP formulation. Our approach achieved significant improvements in terms of the time complexity whilst still able to obtain optimal solutions. In our approach, the value of constant K can be used to control the balance between the optimality of solutions and the computational time.

Chapter 5

Conclusion and Future work

Network survivability is increasingly playing an important role in telecommunications, especially in optical networks on which a tremendous amount of capacity is carried. Survivability in optical networks may be performed at the network design phase in which case it is referred to as pre-configured protection. It may be also be initiated after a failure occurs, known as dynamic restoration. In the context of network optimization and performance, both pre-configured protection and dynamic restoration play important roles in network survivability. The pre-configured protection, or protection in brief, plays a very important role in network planning and optimization, whereas dynamic restoration is the core of the realtime network recovery. In this thesis, we considered pre-configured protection as related to network survivability. Putting it in simple terms, protection is about pre-assigning backup paths for working paths to safeguard against network failures. In protection schemes, backup paths and working paths are set up to optimize an objective function which might target network utilization or network congestion levels. This optimization problem is known as the Survivable Logical Topology Design (SLTD) problem. However, it is critical and imperative to first consider the survivability at the physical layer, without which checking for survivability at the logical layer would be redundant check.

In this thesis, we have investigated network survivability at both physical and logical layers. It is evident that survivability at both of these layers are very important and related to each other. On one hand, if the physical topology is not survivable, then generally speaking, the protection schemes at the logical layer will not be achievable. On the other hand, since protection requirements at the logical layer is application as well as area dependent, in some case, the physical topology may not have to be entirely survivable. The results from our thesis are summarized as follows.

- *Establishing physical survivability accurately and efficiently*:

  Considering that the survivability of the logical topology is heavily dependent on the survivability of the physical topology, establishing the physical survivability of the network is of utmost importance. However, we realized that existing techniques were not able to establish the physical survivability of a moderate size network in a reasonable amount of time. For instance, the cutset technique is not applicable to a network which has more than 30 nodes. In this thesis, we provided a novel theoretical framework for the assessment of the physical survivability of the network. Our proposed algorithms based the framework can cope with large size networks, even in the order of many thousand nodes. For instance, the computational time of the algorithm for a sparse network of 100 nodes is only around 0.1 seconds and increases to less than 1.8 seconds for a network of 500 nodes. These values are measured for sparse networks in which the average nodal degree is ranges from 2 to 4. In the worst-case, the computational time for fully connected mesh networks of 100 to 500 nodes is still acceptable and remains in the order of seconds.

- *Ability to assess both nodal survivability as well as link survivability*:

  As discussed in the example given in Section 3.4.3, the implementation of our physical survivability framework can clearly identify the weaknesses of a network. For the first time, it is possible to establish the survivability of the network not only on the basis of link failures, but also with respect to node failures. Furthermore, our solution gives a comprehensive diagnosis of the network and identifies the exact nodes and links which are the weaknesses of the network, making it unsurvivable.

- *Simultaneously reducing the resource requirements and congestion level in the logical topology design*:

  The next contribution of this thesis is in the area of survivable logical topology design (SLTD). One common SLTD objective is minimization of the total number of wavelength channels used. Another objective is to minimize the congestion level in the network. These objectives are treated separately in the literature, that is, only one of them is targeted at a time. In this thesis, for the first time, we introduced an integrated objective function that can combine the two objectives in the optimization problem. A *weighting factor* was introduced to control the balance between these objectives. For example, the network utilization can be given a higher priority by choosing a weighting factor less than $\frac{1}{W+1}$, where W is the maximum number of wavelength channels available. In our model, where we use the integrated objective, the results are significantly improved over other methods which only consider one objective. For instance, the network utilization in the combined objective case is equivalent to the outcome of those techniques which only aim at minimizing the total number of wavelength used, but the congestion level of our technique is much better. Furthermore, although the congestion level achieved in our combined objective case is a little bit higher than those algorithms which only attempt to minimize the congestion level, we achieve a much better network utilization as a result.

- *Obtaining optimal solutions for SLTD efficiently*:
  Although SLTD has been extensively studied in the literature, it still remains a difficult problem and has been proved to be NP-complete. Time complexity and optimality of solutions are two conflicting metrics for assessing the outcome of SLTD optimization, and it is important to find approaches that can balance these. In this thesis, we proposed a two-step approach that combines the time complexity advantages of existing graph algorithms and optimal solution advantages of the ILP formulation. In the first step, a KDPP algorithm was proposed to find the K-disjoint path-pairs between a source node and a destination node of a connections. The second step was an ILP formulation, but the number of variables was significantly reduced, compared to the original ILP formulation. Our approach achieved significant improvements in terms of the time complexity whilst still able to obtain optimal solutions. In our approach, the value of constant K can be used to control the balance between the optimality of solutions and the computational time.

5.1 Future work

Network survivability is an open problem and can always be improved. A number of possible extensions to this research are itemized below.

- A new survivability verification for physical topology has been proposed in this thesis. A logical extension to this research would be to find the most optimum solution to making an unsurvivable network survivable. For instance, when a node-bridge or link-bridge is identified, there are many ways to remedy the problem. However, each solution will have a different implementation cost and a different impact on the traffic handling abilities of the network. Therefore, it is interesting to find parameters by which we can optimize the design of additional links for the purpose of fixing node-bridges and link-bridges in the physical topology.

- In our studies we had made the assumption that wavelength conversion was available at all nodes of the network. Wavelength converters are expensive equipment, therefore another extension to this research would be to optimize the size and placement of wavelength converters according to the network configuration and traffic requirements.

- Another extension to this research is to study the provisioning of traffic connections with a similar integrated objective function, but with due regards to the resulting quality of service experienced by those connections and implementing different quality of service classes.
- Finally, in efforts to resolve the trade off between the optimality of solutions and the computational time, our proposed heuristic approach can be extended to different protection schemes such as path segment protection and p-cycles.

Bibliography

[1] D. Zhou and S. Subramanian, "Survivability in optical networks," *IEEE Networks*, vol. 14, pp. 16–23, November 2000.

[2] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective.* Academic Press, second ed., 2002.

[3] W. D. Grover, *Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking.* Upper Saddle River, NJ 07458: Prentice Hall PTR, 2004.

[4] ITU-T Reccommendation G.872, "Architecture of optical transport network (OTN)," 1999.

[5] G. Maier, A. Pattavina, S. D. Patre, and M. Martinelli, "Optical network survivability: Protection techniques in the WDM layer," *Photonic Network Communications*, vol. 4, no. 3/4, pp. 251–269, 2002.

[6] W. D. Grover, "Case studies of survivable ring, mesh, and mesh-arc hybrid networks," in *Global Telecommunications Conference*, vol. 1, (Orlando, FL, USA), pp. 633–638.

[7] R. Gould, *Graph Theory.* The Benjamin/Cumming Publishing Company, Inc,, 1988.

[8] E. Townsend, *Discrete Mathematics: Applied Combinatorics and Graph Theory.* The Benjamin/Cumming Publishing Company, Inc,, 1987.

[9] M. Sridharan, M. V. Salapaka, and A. K. Somani, "A practical approach to operating survivable WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 34–46.

[10] C. Mauz, "Unified ILP formulation of protection in mesh networks," in *Proceedings of the 7th International Conference on Telecommunication (COMTEL)*, vol. 2, pp. 737–741.

[11] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh network," *Journal of Lightwave Technology*, vol. 21, no. 4, April. 2003.

[12] H. Zang, C. Ou, and B. Mukherjee, "Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under duct-layer constraints.," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, April. 2003.

[13] S. D. Nikolopoulos, A. Pitsillides, and D. Tipper, "Addresing network survivability issues by finding the k-best paths through a trellis graph," *IEEE Infocom*, 1997.

[14] D. Medhi, "Diverse routing for survivability in a fiber-based sparse network," *Proceeding in IEEE International Conference on Communications (ICC'91)*, pp. 0672–0676, 1991.

[15] C. Xin, Y. Ye, S. S. Dixit, and C. Qiao, "A joint lightpath routing approach in survivable optical networks," *Optical Networks Magazine*, vol. 3, no. 3, pp. 13–20, May/June. 2002.

[16] J. Jang, K. Zhu, L. Sahasrabuddhe, S. J. Ben Yoo, and B. Mukherjee, "On the study of routing and wavelength assignment approaches for survivable wavelength-routed WDM mesh newworks," *Optical Network Magazine*, vol. 4, no. 6, Nov/Dec. 2003.

[17] H. Zang, C. Ou, and B. Mukherjee, "Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under shared-risk-link constraints.," *Proceeding on Asia-Pacific Optical and Wireless Communication (APOC 2001) Conference,*, pp. 49–60.

[18] R. Dutta and G. N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Optical Networks*, vol. 1, no. 1, pp. 73–89, May 1999.

[19] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks.," *IEEE JSAC*, vol. 15, no. 5, pp. 903–908, Jun. 1996.

[20] W. N. Grover and J. Doucette, "Topological design of survivable mesh-based transport networks," *Annals of Operations Research*, vol. 106, pp. 79–125, 2001.

[21] D. Deerter and A. Smith, "Heuristic optimization of network design considering all-terminal reliability," *Proceedings of the 1997 Annual reliability and Maintainability Symposium*, pp. 194–199, January 1997.

[22] S. S. Ellis Horowitz and S. Rajasekaran, *Computer Algorithms*. Computer Science Press, 1998.

[23] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, “Introduce to algorithms,” pp. 550–578, McGraw-Hill, 1990.

[24] J. Surballe, “Disjoint paths in a network,,” *Networks*, vol. 4, pp. 125–145, 1974.

[25] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing.* Kluwer Academic Publishers, 1999.

[26] A. Sen, B. Shen, S. Bandyopadhyay, and J. Capone, “Survivability of lightwave networks - path lengths in WDM protection scheme,” *Journal of High Speed Networks*, vol. 10, no. 4, pp. 303–315, 2001.

[27] S. D. Patre, G. Maier, and M. Martinelli, “Design of static WDM mesh networks with dedicated path protection,” *Infocom*, 2002.

[28] D. Xu, Y. Xiong, and C. Quiao, “Novel algorithms for shared segment proteciton,” *IEEE Journal on Selected Areas in Communications*, vol. 21.

[29] G. Shen and W. D. Gover, “Extending the *p*-cycle concept to path segment protection for span and node failure recovery,” *IEEE Journal on Selected Areas in Communication*, vol. 21, no. 8, pp. 1306–1319, 2003.

[30] D. Schupke, C. Gruber, and A. Autenrieth, “Optimal configuration of p-cycles in WDM networks,” in *IEEE International Conference on Communications (ICC 2002)*, vol. 5, pp. 2761- 2765.

[31] W. Grover and J. Doucette, “Advances in optical network design with p-cycles: Joint optimization and pre-selection of candidate p-cycles,” *Pro. of IEEE/LEOS Summer Topicals 2002*, pp. 49–50, July 2002.

[32] C. Mauz, “p-cycles protection in wavelength routed networks,” *Proceeding of the Seventh Working Conference on Optical Network Design and Modelling (ONDM’03)*, Feb 2003.

[33] S. Ramamurthy and K. Sivarajan, “Routing and wavelength assignment in all-optical networks,” *IEEE/ACM Trans. Networking*, vol. 3, pp. 489–500.

[34] D. Eppstein, “Finding the *k* shortest paths,” *Society for Industrial and Applied Mathematics (SIAM J. COMPUT.)*, vol. 28, no. 2, pp. 652–673, 1998.

[35] E. D. Q. V. Martins and J. L. E. D. Santos, “A new shortest paths ranking algorithm,” *Investigao Operacional*, vol. 20, no. 1, pp. 47–62, 2000.

[36] E. D. Q. V. Martins, M. M. B. Pascoal, and J. L. E. D. Santos, “The *k* shortest paths problem,” tech. rep., CISUC, Jun. 1998.

[37] J. A. d. Azevedo, J. J. E. R. S. Madeira, E. Q. V. Martins, and F. M. A. Pires, "A shortest paths ranking algorithm," *Proceedings of the Annual Conference AIRO'90 Operational Research Society of Italy*, pp. 1001–1011, 1990.

LIST OF PUBLICATIONS

1. Viet Q. Phung, Daryoush Habibi, Hoang Nghia Nguyen and Kungmeng Lo. "An efficient approach to optimal wavelength routing in WDM optical networks". *Proceeding 12th IEEE international conference on networks,* Vol. 2, pp. 600-604, December, 2004.

2. Daryoush Habibi, Viet Q. Phung and Hoang Nghia Nguyen. "A Tractable Wavelength Routing Solution for Optical Networks". *Australian Telecommunication Networks and Applications Conference, ATNAC 2004 (Network).*

3. Kungmeng Lo, Daryoush Habibi, Viet Q. Phung. "Lens design issues for the optical structure of a holographic free space optical switch". *Proceedings of SPIE 5276, 365 (2004)*