

2012

# A Test-Bed for Comparing Impairment Aware Routing & Wavelength Assignment Algorithms in WDM Networks

Shrestharth Ghosh

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

---

## Recommended Citation

Ghosh, Shrestharth, "A Test-Bed for Comparing Impairment Aware Routing & Wavelength Assignment Algorithms in WDM Networks" (2012). *Electronic Theses and Dissertations*. 4807.  
<https://scholar.uwindsor.ca/etd/4807>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email ([scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca)) or by telephone at 519-253-3000ext. 3208.

A Test-Bed for Comparing Impairment Aware Routing &  
Wavelength Assignment Algorithms in WDM Networks

by  
Shrestharth Ghosh

A Thesis  
Submitted to the Faculty of Graduate Studies  
through Computer Science  
in Partial Fulfillment of the Requirements for  
the Degree of Master of Science at the  
University of Windsor

Windsor, Ontario, Canada

2012

© 2012 Shrestharth Ghosh

**A TEST-BED FOR COMPARING IMPAIRMENT AWARE-  
ROUTING WAVELENGTH ASSIGNMENT ALGORITHMS  
IN WDM NETWORKS**

by  
Shrestharth Ghosh

APPROVED BY:

---

Dr. Ronald Barron, External Reader  
Department of Mathematics and Statistics

---

Dr. Arunita Jaekel, Internal Reader  
School of Computer Science

---

Dr. Subir Bandyopadhyay, Advisor  
School of Computer Science

---

Dr. Luis Rueda, Chair of Defence  
School of Computer Science

July, 2012

## **AUTHOR'S DECLARATION OF ORIGINALITY**

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

## ABSTRACT

When an optical signal propagates through optical fibers, the quality of the signal degrades due to a number of physical phenomena. Traditional Routing and Wavelength Assignment (RWA) approaches assume an ideal physical layer medium and ignore the effects of physical layer impairments on the lightpath feasibility. In the last few years investigators have started taking into account the fact that the quality of transmission (QoT) of an optical signal propagating through an optical network degrades, due to physical layer considerations. To measure the extent of this degradation due to physical layer impairments (PLI), metrics such as the Bit Error Rate (BER) used. In a translucent network, when the quality of a signal is reduced sufficiently, the signal has to be regenerated. In a transparent network, regenerators are not allowed so that lightpaths with high bit error rates are disallowed. A number of heuristic approaches for impairment aware RWA have been proposed for transparent and for translucent networks.

As a result of this investigation a test-bed has been developed for Impairment Aware Static Route and Wavelength Assignment (IA-RWA) in transparent networks. This includes a tool for computing BER values and allows the user to run a new heuristic for IA-RWA and study its performance against a number of existing heuristics for IA-RWA.

# DEDICATION

*To my dearest lil' sis.*

## ACKNOWLEDGMENT

I would like to take this opportunity to convey my sincere appreciation to my supervisor, Dr. Subir Bandyopadhyay for his continuous guidance all throughout my graduate studies. This work could not have achieved its completeness without his constant encouragements, valuable advices and proper guidance. I would also like to thank him for his immense amount of patience that he showed while working with me.

I would like to thank all the committee members, Dr. Ron Barron, Dr. Arunita Jaekel and Dr. Luis Rueda for their valuable time, suggestions and constructive comments.

I would also like to thank my colleagues, Sebastian Zawada, Sriharsha Varanasi and Frank Luo for their valuable inputs.

This acknowledgment would remain incomplete without thanking Naseer Ansari and his family and a good friend Satish Panigrahi for their continuous support and encouragements.

Shrestharth Ghosh, Windsor

# CONTENTS

<b>AUTHOR'S DECLARATION OF ORIGINALITY</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>DEDICATION</b>	<b>v</b>
<b>ACKNOWLEDGMENT</b>	<b>vi</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF TABLES</b>	<b>xii</b>
<b>LIST OF SYMBOLS</b>	<b>xiii</b>
<b>Chapter 1 INTRODUCTION</b>	<b>1</b>
1.1 Overview	1
1.2 Motivation	3
1.3 Significance of the Work	4
1.4 Organization of Thesis	4
<b>Chapter 2 REVIEW OF RELATED TECHNOLOGY</b>	<b>5</b>
2.1 Overview	5
2.2 Components of Optical Network	5
2.2.1 Optical Fibers	5
2.2.2 Wavelength Routers and Optical Add Drop Multiplexers	9
2.2.3 Optical Transmitters, Modulators, and Receivers	12
2.2.4 Optical Signal Amplifiers & Regenerators	13
2.3 Fundamentals of Optical Networks	15
2.3.1 Wavelength Division Multiplexing & WDM Network	15
2.3.2 Optical Network Representation	17
2.3.3 Routing and Wavelength Assignment	18



2.3.4	Types of Optical Networks	21
2.3.5	Physical Layer Impairment	24
2.3.6	Classification of Physical Impairment	25
2.4	$k$ -Shortest Path Algorithms	27
2.4.1	Importance of $k$ -Shortest Path Algorithms	27
2.4.2	Yen's Algorithm	27
2.5	Relevant Work Done in IA-RWA	29
2.5.1	Approaches to PLI-RWA	30
2.5.2	Previous Work	30
<b>Chapter 3</b>	<b>DESIGN AND IMPLEMENTATION OF THE TEST-BED</b>	<b>37</b>
3.1	Overview	37
3.2	The Test-bed	41
3.2.1	The Simulator	42
3.2.2	The Dataset Generator	44
3.2.3	The Network Topology Generator	45
3.2.4	The $k$ -Shortest Path Algorithm	45
3.3	Implementation of the Heuristic	55
3.3.1	Classical RWA (cRWA) with $k$ -Shortest Paths	56
3.3.2	sLERP	58
3.3.3	Shortest (Longest) Path First	61
<b>Chapter 4</b>	<b>EXPERIMENTAL ANALYSIS</b>	<b>64</b>
4.1	Overview	64
4.2	Synthetic Networks	66
4.2.1	Channels per Fiber ( $l$ )	67
4.2.2	Number of Nodes ( $N$ )	68
4.2.3	Demands ( $D$ )	69
4.2.4	Edge Length	71
4.2.5	Minimum Number ( $K$ ) of Shortest Paths between a Source-Destination Pair	72
4.2.6	$k$ Shortest Paths (K_PATHS)	73
4.3	Real-life Networks	74
4.3.1	American Topology	74
4.3.1.1	Channels per Fiber ( $l$ )	75
4.3.1.2	Range for the Shortest Path	76
4.3.1.3	Demands ( $D$ )	79
4.3.1.4	Minimum Number ( $K$ ) of Shortest Paths between a Source-Destination Pair	79
4.3.2	USANET	80

4.3.2.1	Channels per Fiber ( $l$ )	80
4.3.2.2	Demands ( $D$ )	81
4.3.2.3	Minimum Number ( $K$ ) of Shortest Paths between a Source-Destination Pair	83
4.3.3	ARPANET	83
4.3.3.1	Channels per Fiber ( $l$ )	84
4.3.3.2	Demands ( $D$ )	85
<b>Chapter 5</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>87</b>
5.1	Conclusions	87
5.2	Future Work	88
	<b>REFERENCES</b>	<b>89</b>
	<b>VITA AUCTORIS</b>	<b>97</b>

## LIST OF FIGURES

Figure No	Title	Page
2.1	Numerous optical fibers bundled together	6
2.2	Optical fiber cross-section	7
2.3	Total internal reflection (TIR)	8
2.4	An optical signal reaches its destination through numerous Total Internal Reflection	8
2.5	Static optical cross-connect switch	9
2.6	Dynamic optical cross-connect switch	10
2.7	Schematic diagram for an Optical add-drop multiplexer (OADM)	11
2.8	The model of a regenerator and the three major functions, optical receiver, electronic amplifier, and optical transmitter	15
2.9	Channel spacing	16
2.10	A graph representing the network of four networking device communicating	17
2.11	The physical topology of a typical WDM network with four end nodes $E_1, \dots, E_4$ and four routers $R_1, \dots, R_4$	18
2.12	Logical topology corresponding to the lightpaths shown in Figure 2.10	19
2.13	Demand set categories	20
2.14	A simple network	23
2.15	At Node $E_4$ a Regenerator $R_1$ is placed, as the distance of the path from $E_1$ to $E_3$ is greater than optical reach	24
2.16	Various PLI-RWA approaches	31
3.1	Block diagram for the “Simulator”	38
3.2	A simple network	40
3.3	Simulator interaction diagram	43
3.4	An implementation of $k$ -SP	46
3.5	An example network	47

3.6	$A$ 's direct neighbors are $B, G$ and $D$ . Hence these are marked "reachable nodes"	48
3.7	Different transitions a graph (or a network) goes through in Dijkstra's shortest path algorithm	49
3.8	The final graph	50
3.9	<i>min</i> -Heap example and its array implementation	51
3.10	An 10-node network	52
3.11	Different steps in the $k$ -SP	53
3.12	Example topology used for illustration of the heuristics	55
4.1	Synthetic network behavior with varying $\lambda$	67
4.2	Varying $N$ with $\lambda = 8$	68
4.3	Varying $N$ with $\lambda = 16$	69
4.4	Varying $D$ with $\lambda = 8$	70
4.5	Varying $D$ with $\lambda = 16$	70
4.6	Varying Edge Length Lower Limit-Upper Limit with $\lambda = 8$	71
4.7	Varying Edge Length Lower Limit-Upper Limit with $\lambda = 16$	72
4.8	Success % for varying $K$ values	73
4.9	Varying $K\_PATHS$	74
4.10	American Topology	75
4.11	Effect of varying $\lambda$ in AT	76
4.12	Effect of varying $PERCENTAGE\_UPPER\_BOUND$ ( $pub$ ) and $PERCENTAGE\_LOWER\_BOUND$ ( $plb$ ) in AT for $\lambda = 8$	77
4.13	Effect of varying $PERCENTAGE\_UPPER\_BOUND$ ( $pub$ ) and $PERCENTAGE\_LOWER\_BOUND$ ( $plb$ ) in AT for $\lambda = 16$	78
4.14	Effect of varying $PERCENTAGE\_UPPER\_BOUND$ ( $pub$ ) and $PERCENTAGE\_LOWER\_BOUND$ ( $plb$ ) in AT for $\lambda = 16$	78
4.15	Effect of Varying $D$ in AT	79
4.16	Effect of Varying $D$ in AT	80
4.17	USANET	81
4.18	Effect of varying $\lambda$ in USANET	81
4.19	Effect of varying $D$ USANET	82
4.20	Average Success rate for varying $D$ in USANET	83
4.22	ARPANET	83
4.21	Effect of varying $K$ values in USANET	84
4.23	Effect of varying $\lambda$ in ARPANET	85
4.24	Effect of varying $D$ ARPANET	85

## LIST OF TABLES

Table No	Title	Page
2.1	Different approaches to comparisons	33
3.1	The “Candidate” shortest paths. <i>NF</i> =Not Found	55
3.2	Demands with corresponding <i>k</i> -SPs	57
3.3	Summary of channel assignment phase.	58
3.4	Demands with corresponding <i>k</i> -SPs	59
3.5	Summary of the permutation and wavelength assignment	61
4.1	Features of real networks	64
4.2	Some well known parameters	65
4.3	OSNR constants	65
4.4	Constant parameters	66

## LIST OF SYMBOLS

ALD	ad hoc Lighpath Demands
BER	Bit Error Rate
DLD	Dynamic Lighpath Demands
DWDM	Dense Wavelength Division Multiplexing
FSK	Frequency Shift Keying
Gbps	Giga bits per second
GHz	Giga hertz
IM/DD	Amplitude Direct Detection
Mbps	Mega-bits per seconds
nm	nano meter
OC	Optical Channel transmission rates
OEO	Optcal-Electronic-Optical
OOK	On/Off Keying
OON	Opaque Optical Network
OPT	OSNR based PLI Tool
OSNR	Optical Signal-to-Noise Ratio
OXC	Cross Connect Switch
PLD	Permanent Lighpath Demands
PLI	Physical Layer Impairments
PSK	Phase Shift Keying
QoT	Quality of Transmission
RDG	Random Datset Generator
RNG	Random Network Generator
RWA	Routing and Wavelength Assignment
SLD	Scheduled Lighpath Demands
Tbps	Terabits per second
TON	Transparent Optical Network
WDM	Wavelength Division Multiplexing

# Chapter 1

## INTRODUCTION

### 1.1 Overview

Telecommunication is the discipline of communicating over some distance using telephone or radio technology [1]. To achieve such communication, components like microelectronic computers and PC technologies are used [1]. With the help of these components it is now possible to transmit and receive voice, data and video communications [1]. Internet, as we know it, is a world-wide interconnection of computer networks that allows the users at any computer system on the network to access information from another computer [1] on the network. These systems of computer networks are connected through different types of transmission mediums that allow the computers to transmit the data in various forms (electrical, optical, microwave, etc.). A wide range of such transmission media exists. Conventionally, these types of media are widely divided into two major categories – **Guided Media** and **Unguided Media**. Guided media involves cables and optical fibers and unguided media involves microwaves, infrared, Bluetooth, and Wi-Fi.

The connection between the networking devices are generally categorized based on the distances between them. A *Local Area Network* (LAN) is a network which has the capability of distributing the data around a single facility or a campus [1]. A *Metropolitan Area Network* (MAN) is a network that connects multiple such LAN's in a relatively localized geographical area. Enterprise networks have the capability to support multiple such clusters of LAN and hence known as *Wide Area Network(s)* (WAN) [1]. WAN are large networks that can span across the globe. Using such interconnections, everyone in a business or government organization can communicate with one another from every accessible facility on the network.

The *bandwidth* (throughput) of a network is defined as the amount of

information one can transmit over a transmission medium [1]. The exponential growth of internet traffic, due to high-speed applications such as video conferencing, high quality video on-demand, etc., is imposing a huge demand for bandwidth on the telecommunication infrastructure [2]. The internet is maintained by various collaborating Internet Service Provider (ISPs). Every ISP has the responsibility of keeping their network operational, while interfacing with other ISPs using a set of well-known networking protocols and standards. It was realized that replacing electrical signals with optical signals can increase the capacity of a network by several orders of magnitude [3]. When an underlying telecommunication infrastructure uses optical signals over the optical fibers for data communication, there are numerous benefits such as huge transmission bandwidth, low signal corruption, low power requirement, low cost, etc. [2]. In any communication system, data is generally available as an electrical signal that may take the form of an analog signal or digital signal. In the case of analog, the signal changes continuously with time [3]. On the contrary, the digital signal normally takes only two distinct values, binary 1 and 0 [3]. To use an optical network, data in the form of electrical signals has to be converted to optical signals [3]. To achieve this, the optical signal at the source is modulated by applying the electrical data to be communicated.

The throughput of an optical fiber is too high to be completely used by a single optical signal with one carrier wavelength, due to speed limitation in the electrical devices in a computer. *Wavelength Division Multiplexing* (WDM) is the technology of communicating more than one carrier wavelength on a single optical fiber. A wavelength needs to be assigned to the path between a source and a destination, so that the source can send data to the destination on a specific carrier wavelength. This issue is termed as *Routing and Wavelength Assignment* (RWA) problem. A *lightpath* is a connection at optical layer, which starts at the source and propagates through optical and various networking components to reach the destination. In cases where wavelength conversion is not available, a lightpath must be assigned a single wavelength on all fibers in its path. This constraint is widely known as the *Wavelength Continuity Constraint*. Any algorithm trying to solve the RWA problem needs to consider this constraint.

Though the introduction of approaches like WDM and RWA solved the issue of bandwidth allocation to some extent, another issue still plagues the optical networks. While communicating encoded data on an optical signal over a fiber, due to various physical phenomena, different types of signal degradation in the form of noise and distortions are introduced into the signal. This kind of “pollution“ of the optical signal leads to introduction of errors into the optical signal. When the optical signal reaches its destination, errors can be significant



enough for the destination to interpret the data incorrectly. To address the issue of impairments (errors) in optical networks, researchers have proposed many solutions that consider these impairments while solving the RWA problem. This added issue of impairments along with wavelength continuity constraint adds further complications while solving the RWA issue. Such RWA solutions are known as *Impairment Aware Routing and Wavelength Assignment* (IA-RWA). To quantify these impairments, authors have used the notion of Bit Error Rate (BER) and Optical Signal to Noise Ratio (OSNR) to measure the extent of degradation. These metrics can give an approximate measure of the impairments in the network and also are useful in determining the threshold up to which the errors are acceptable.

Developing this test-bed required a lot of planning and testing. An *iterative software development* approach was considered when dealing with this approach. The main idea behind such approach is to embrace the change in the requirements all through the process of software development. Regular meetings and discussions allowed the development of better *use cases*, which essentially are the short stories that describe some of the major functionalities of the system. In Chapter 3, the interaction between different software components that constitute this test-bed are described in detail. To design these interactions, IBM provides an intelligent tool called Rational Rhapsody, which was used in this thesis. During the development phase of the project, Eclipse CDT, a well known Integrated Development Environment (IDE), was used to develop C/C++ code. To check different memory leaks and test different scenarios, an open source tool called Valgrind was used.

## 1.2 Motivation

As indicated in Section 1.1, due to the use of the WDM technology, an optical signal traversing through an optical fiber has to share the optical fiber with numerous other optical signals of different wavelengths. This may introduce a significant amount of noise and disturbance among the optical signals which may lead to erroneous data. Time and again many authors have come up with different approaches to deal with the issue of such impairments. It was observed that the comparisons produced in the literature lacked a standard test tool that can efficiently compare performance of the new heuristics with the performance of the existing ones. The lack of such tool acts as the motivation for the work done in this thesis.

### **1.3 Significance of the Work**

As the field of optical networking evolves, better heuristics will be proposed to deal with the issue of impairment in optical networks. Any new work generally has to be evaluated by comparing it to the past work done in this area of research. To the best of our knowledge no researcher has developed a common platform for comparing different heuristics for IA-RWA. In the case of IA-RWA, a set of source and destination pairs is part of the input to a heuristic. The pairs of source and destination nodes essentially represents the requests for communication between the pairs, hence known as demands. The performance of a heuristic for IA-RWA may be measured based on the number of demands that were successfully established by the heuristic. In this thesis, a standard test-bed has been implemented to enable users to compare the performance a new heuristic for IA-RWA to the existing ones. The test-bed provides an interface through which a variable number of heuristics to solve the IA-RWA issue can be plugged in. The test-bed will have the capability of supplying the heuristics with the appropriate set of inputs necessary for execution of the heuristics. For the purpose of measuring the extent of impairments, an external tool (OSNR based PLI Tool) was also integrated with the test-bed. For better analysis of heuristics, the test-bed allows the user to tweak various parameters such as the number of nodes in a network, length of the edges, etc. The test-bed also includes many real-life network topologies, which gives the opportunity to analyze the behavior of the heuristics in real-life situations.

### **1.4 Organization of Thesis**

The remainder of the thesis is organized as follows. In Chapter 2 relevant topics in WDM networks are reviewed and some papers dealing with impairments in optical networks are summarized. In Chapter 3 the working and the implementation of the test-bed and its components are discussed in detail. This chapter also discusses the heuristics that have been implemented to test the credibility of the test-bed. In Chapter 4 a summary of different kinds of experimental testing and analysis is given. Finally, Chapter 5 concludes the work done in this thesis and also suggests some possible future works.

## Chapter 2

# REVIEW OF RELATED TECHNOLOGY

### 2.1 Overview

This chapter presents a brief introduction to optical networks and gives a comprehensive literature review of the work done in Routing and Wavelength Assignment, while considering Physical Layer Impairments. The objective of this chapter is to allow the reader, who may not be familiar with WDM networks, sufficient background to follow the work done in this thesis. This chapter introduces the basic components of optical networks and gives an overview of the terms and concepts used in this topic. Subsequent sections will describe the basic operations of WDM optical networks and review the work related to the research reported in this thesis.

### 2.2 Components of Optical Network

In an optical network various components communicate with each other, following some optical layer protocols, in addition to the general protocols followed in general networking [4]. To facilitate data communication between various components, the optical network must be equipped with facilities for generating optical signals from the electrical information that the optical components receive from electronic components[4]. This section will give further details about the various components that actively or passively take part in the data communication in optical networks.

#### 2.2.1 Optical Fibers

Optical fibers constitute the medium for communication in an optical network, carry light signals across the network and allow the various components to communicate with each other. Optical fibers are extremely thin and may be visualized

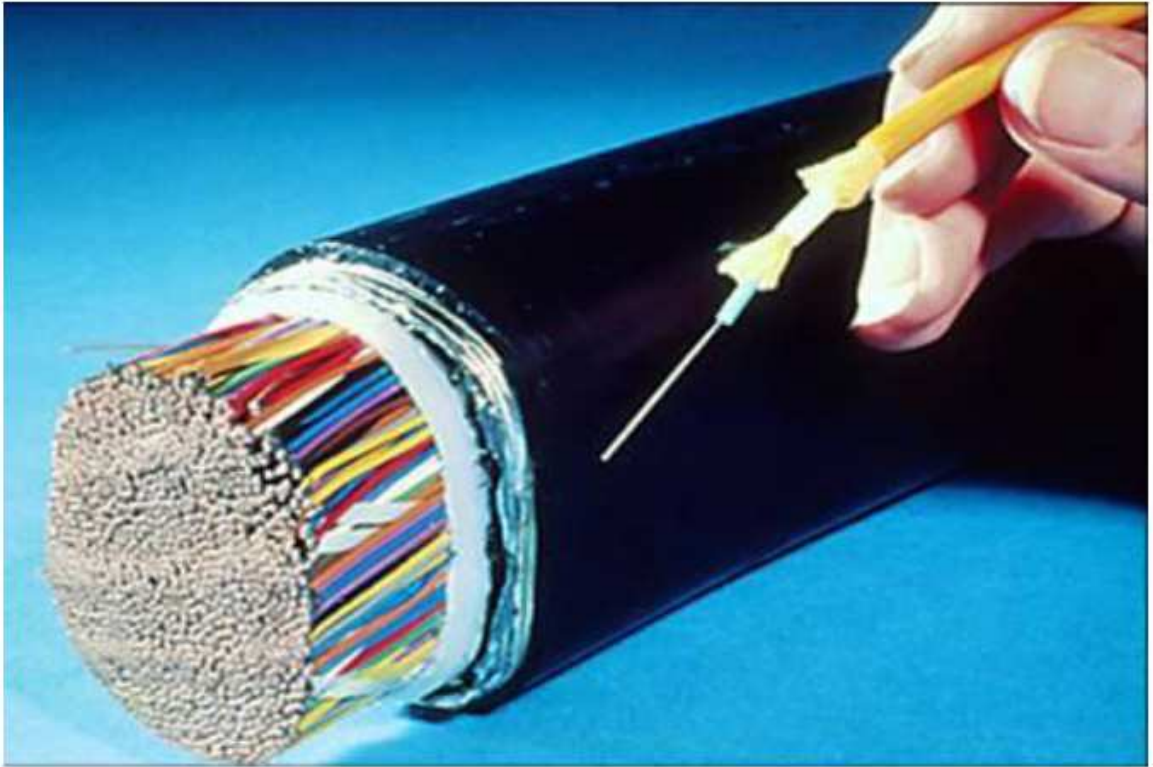


Figure 2.1: Numerous optical fibers bundled together

as long glass-pipes, which are also called *filaments*. An optical fiber typically has a diameter comparable to that of a human hair and can transmit data ranging from 2.5 Gbps to 40 Gbps over a long distance (thousands of kilometers).

Due to the fragility of the filaments, optical fibers must be protected from mechanical and other damages and are buried underground. Figure 2.2, from [5], shows the dimensions of an optical fiber. The accompanying diagram (Figure 2.1 [5]) shows how multiple optical fibers can be bundled up together to form a single large cable. The central part of the optical fiber is the core; the size of the core can be  $8\ \mu\text{m}$  (mono-way),  $50\ \mu\text{m}$  (multi-way) or  $62.5\ \mu\text{m}$  (multi-way). The coating can have a diameter of  $125\ \mu\text{m}$  [5]. For the purpose of extra durability and strength, the cables are covered by an armored deck, which is semi-rigid and does a good job of protecting the nucleus (core).

The transmission of optical signals along the optical fiber is based on various laws of refraction. Refraction of light occurs when the light experiences a change in its speed [6] as shown in Figure 2.3. When light traveling through a medium having a refractive index of  $\mu_1$  meets a second medium, having a refractive index of  $\mu_2$  ( $\mu_1 > \mu_2$ ), at an angle greater than the *critical angle*  $\sin^{-1}\left(\frac{\mu_2}{\mu_1}\right)$  [4], *total internal reflection* takes place. The change in refractive indices is experienced by a light signal when it travels from one optical medium to another, and if the refractive index of the former optical medium is larger than the later,

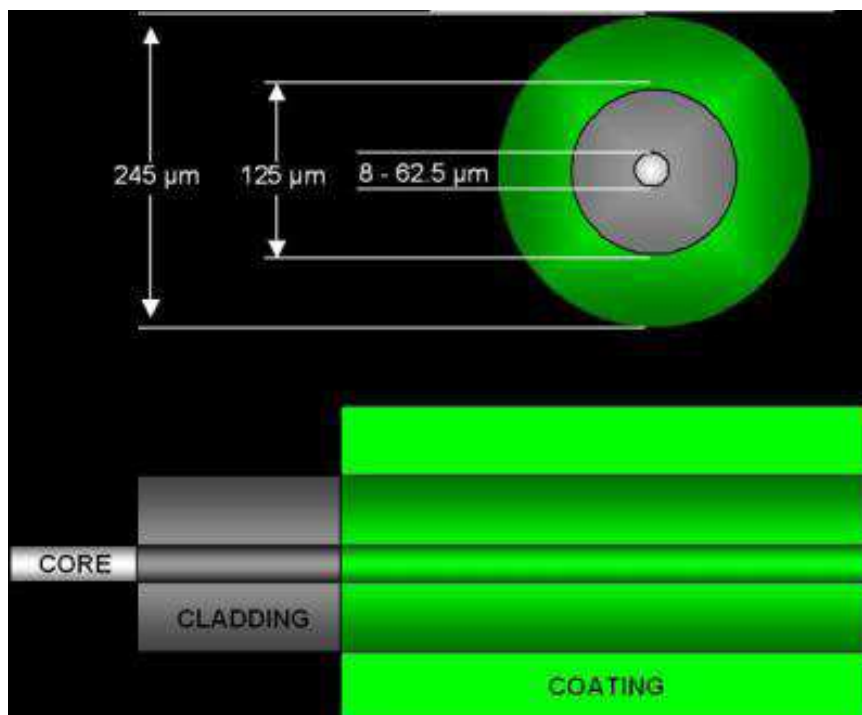


Figure 2.2: Optical fiber cross-section

a total internal reflection may occur if the light enters the medium at an angle exceeding the critical angle. In Figure 2.3,  $n_1$  and  $n_2$  are the two different optical medium, an optical signal at different angles is represented by the two different lines (dashed and simple). The critical angle is represented by the angle formed by the dashed line which essentially exhibits the notion of total internal reflection. As shown in Figure 2.4 (from [7]), optical signals propagate through fibers in optical networks using total internal reflections [4].

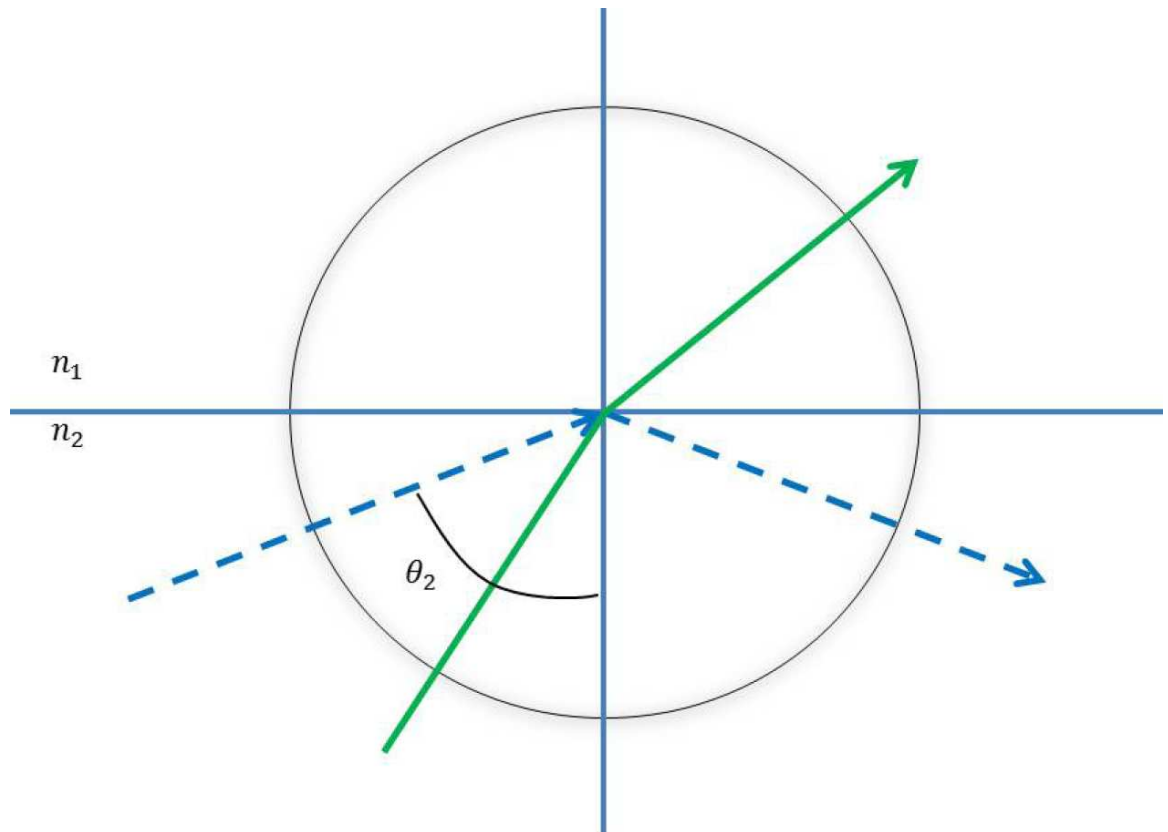


Figure 2.3: Total internal reflection (TIR)

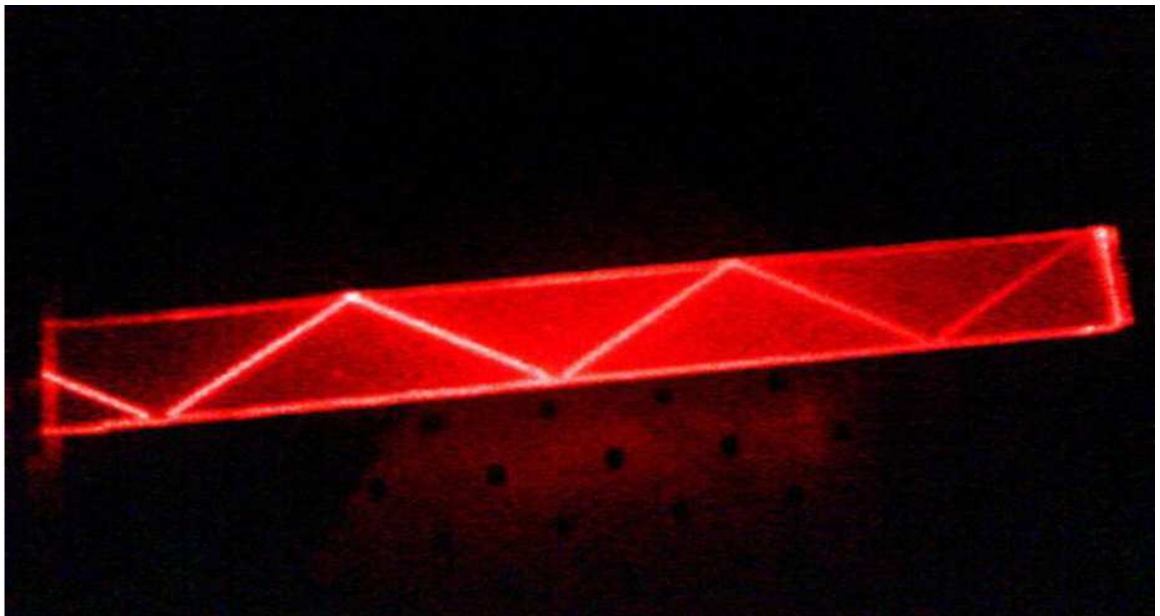


Figure 2.4: An optical signal reaches its destination through numerous Total Internal Reflection

### 2.2.2 Wavelength Routers and Optical Add Drop Multiplexers

Wavelength routers or Lambda routers or Optical cross-connects (OXC) are devices that switch each incoming optical signal to some desired port in the router. This type of intelligent switching of optical networks finally results in route selection for an optical signal. An OXC, at a minimum, consists of a multiplexer and a de-multiplexer with each output of the de-multiplexer connected to the input of the multiplexer. The connected input and output ports of the pair are responsible for routing and relaying the optical signals. If the connections between the outputs and inputs of the multiplexer-de-multiplexer pair are fixed, then the OXCs are called *static* OXCs. If there is flexibility in choosing the multiplexer inputs, then the OXC is termed as *dynamic* OXC. Some features for an ideal optical cross-connect switch, are as follows [8]:

1. Scalability;
2. High-port-count switches;
3. The ability to switch with high reliability, low loss, good uniformity of optical signals independent of path length;
4. The ability to switch to a specific optical path without disrupting the other optical paths.

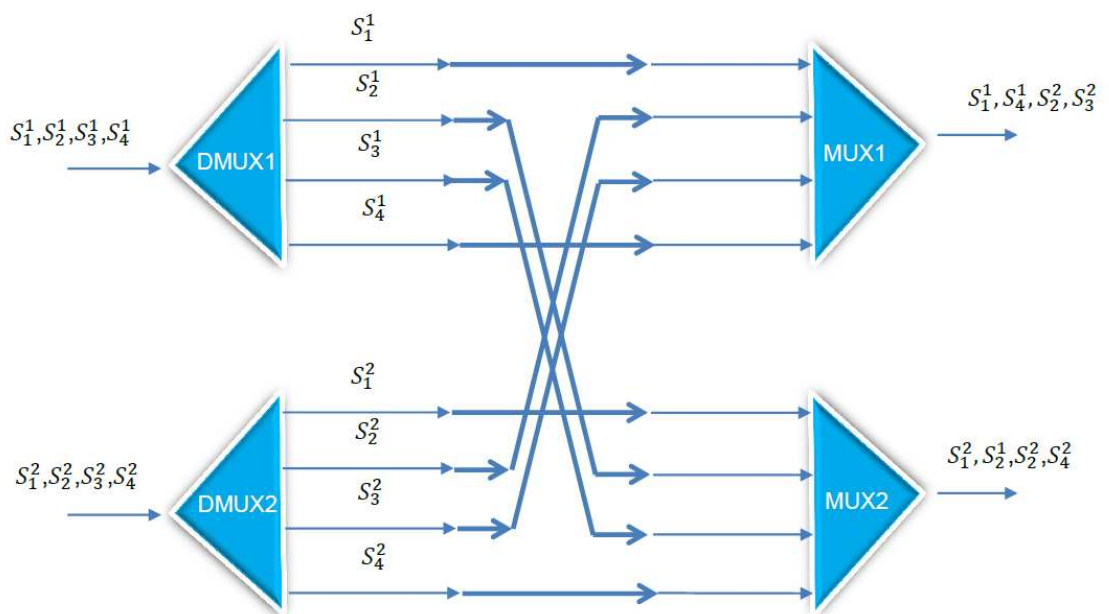


Figure 2.5: Static optical cross-connect switch

Figure 2.5, taken from [4], gives an illustrative diagram of a static optical cross-connect switch, where the connections between the outputs of the de-multiplexer and inputs of the multiplexer are fixed. When a network uses *static* OXCs to route optical signals, then the routing will be fixed, as an output port of a de-multiplexer will always direct an optical signal to a fixed input port of the multiplexer. As the routing is fixed, the routing decisions made by the algorithm are evaluated during the network set-up phase. Figure 2.6 (from [4]) represents a *dynamic* cross-connect switch that allows the flexibility of changing the connections between the outputs of the de-multiplexer and inputs of the multiplexers. To realize this kind of flexibility the OXC also includes a *switching fabric* in its core [9] which essentially is the combination of hardware and software components that moves the data coming in to an optical switch out by the correct port to the next node in the network. Apart from dynamic and static type of OXCs, wavelength routers are also classified based on the network domain under which they work. When an OXC does not have the capability to switch the optical signals in the optical domain, then the cross-connect switch performs an operation, widely known as *O-E-O conversion*, in which the device converts the optical signals into electrical signals [8]. Once the switching is done, the electronic signal is again switched back to an optical signal, for further communication. Another type of OXC is called *all-optical cross-connect*, that can switch the data without any O-E-O conversions. The cores of such OXCs are equipped with an optical switch which is independent of data rate and various network layer protocols; this feature allows the OXCs to be ready for any future data-rate upgrades [8].

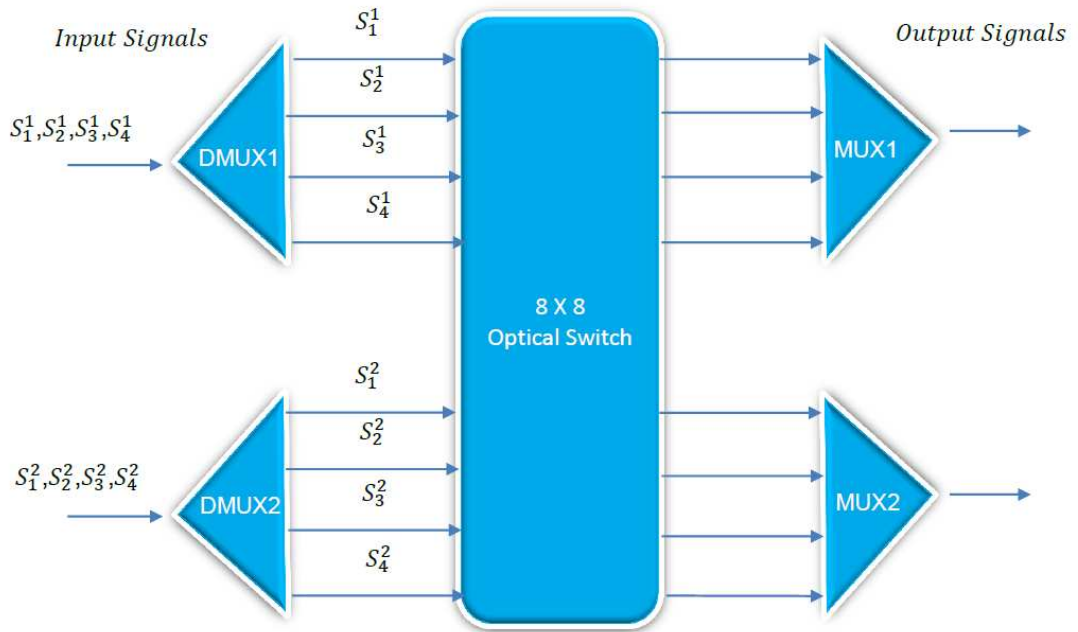


Figure 2.6: Dynamic optical cross-connect switch



An Optical Add-Drop Multiplexer (OADM) is one of the core components for optical communication technology. OADM has the main responsibility to route, add and drop optical signals. OADMs provide a cost effective mechanism to handle the traffic for both metro and long-haul networks [10]. The input and output ports of OADM are essentially connected to optical fibers. The core that allows an OADMs to switch optical signals is an OXC. OADMs have the additional capability of “dropping” optical signals at a de-multiplexer end or “adding” optical signals at a multiplexer end. These two attributes are achieved by connecting a few outputs of the de-multiplexer to a receiver and a few inputs of the multiplexer to a modulator/transponder.

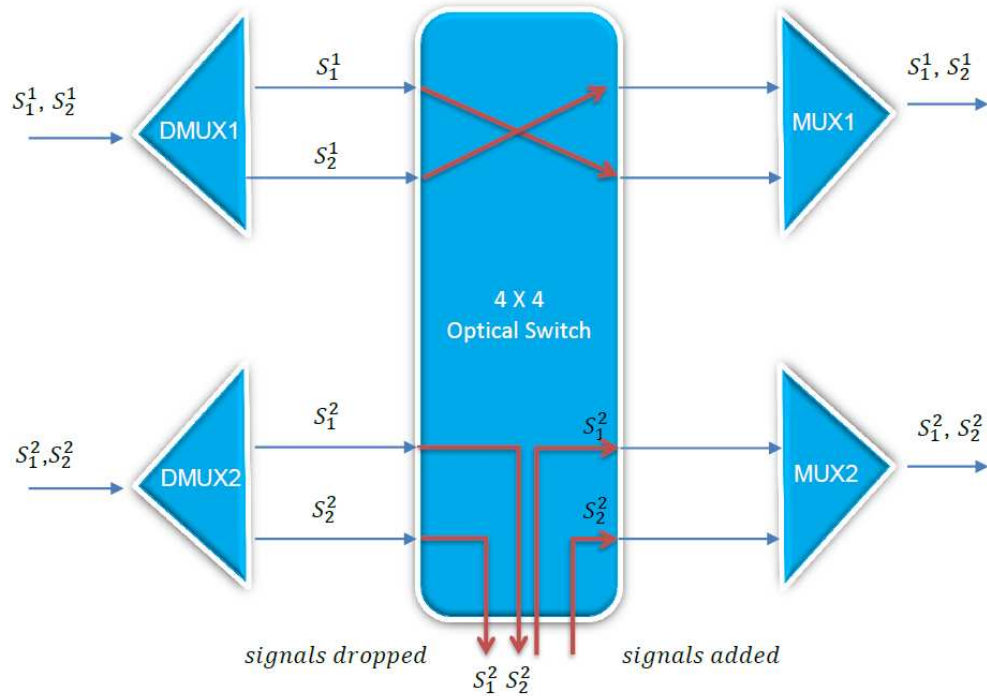


Figure 2.7: Schematic diagram for an Optical add-drop multiplexer (OADM)

Figure 2.7 (taken from [4]) shows the mechanism of an OADM, where there are two pairs of de-multiplexers and multiplexers and a switching fabric. As outlined above, a few chosen outputs of the de-multiplexer are connected to the selected few inputs of the multiplexer. In the figure the optical signals  $s_1^1, s_2^1$  use the input port of *DMUX1* to get routed to output port of *MUX1*. Here the signal pair  $s_1^2, s_2^2$  traversing the input port of *DMUX2*, are dropped by the de-multiplexer. Signals may have reached the destination, and after they are dropped, the signals will undergo O-E-O conversion for the use of the end node containing this OADM. The figure also shows that the end node generates

two optical signals which are added to the network by the connecting modulator/transponder of the end node to the multiplexer. The OADMs, when used with optical switches, are considered to be reconfigurable, in terms of choosing a particular wavelength that needs to be dropped/added from/to the fiber.

### 2.2.3 Optical Transmitters, Modulators, and Receivers

As the name suggests, an optical transmitter is a device used to generate optical signals for communication over an optical fiber. The transmission of an optical signal is the combined result of various steps that the transmitter has to consider while transmitting the optical signal. With the help of multiple transmitters, different electrical signals carrying different data can be transmitted over a single optical fiber, using variable number of carrier wavelengths. This technique is called *Wavelength Division Multiplexing* (refer to Section 2.3.1 for a comprehensive reading). The transmitter converts the electrical signal into an optical signal [6]. There are two major types of optical signal generators, namely, *Light-Emitting Diodes* (LEDs) and costly and high power *Light Amplification by Stimulated Emission of Radiation* (Laser). Lasers are considered to be eminently important optical signal generators that can generate an optical signal which has an important property that allows all photons to be in the same phase [4]. Some literature have negatively criticized the usage of lasers as the sources of optical signal generation, e.g. in [11] authors point out that due to non-uniformity in various properties of the laser beam, such as polarization, output power of the modulated laser beam etc., the usage of such technologies should be restricted. It is also pointed in [4] that although the fixed wavelength transmitters are capable of overcoming the drawbacks of the laser technology, since the optical networks nowadays carry large numbers of different wavelengths, and the concept of one transmitter per optical signal is proving to be expensive.

The modulation of an optical signal is required for sending data using an optical fiber. Modulation is the process of converting the data in an electronic format to optical format [4]. In general, during the modulation process, data are encoded by modifying a carrier wavelength. A modulated laser beam can carry data at a current commercially available data rate of up to 40 Gbps. There are varying approaches to optical modulation; all of them are categorized based on the property of the laser beam used in modulation [12]. When the *phase* of the light beam is modulated the method is called *phase shift keying* (PSK); when the *state of polarization* is modulated it is called the *state-of-polarization shift keying* (SoPSK); when the property to be modulated is *frequency*, it is called *frequency shift keying* (FSK); finally in the case of *amplitude modulation*, it is categorized as *amplitude shift keying* (ASK) [12], which further has two types – intensity

modulation with *amplitude direct detection* (IM/DD) and *on-off keying* (OOK) modulation. Among the two (IM/DD and OOK), the most common scheme is the latter where the light off (on) represents the bit 0 (1) [4].

The main role assumed by the receiver is to detect the modulated photonic signal with a certain level of accuracy. The photonic (or optical) signal originates at a transmitter in the source node, travels through an optical fiber for long distances to finally arrive at the receiver in the destination. To detect the optical signal with high accuracy, an optical receiver is generally equipped with a powerful and quick-responding photo-detector, an electronic filter and an optical preamplifier [11]. The author of [13] list some of the important requirement a receiver must fulfill –

1. A photo-detector or optical receiver should be highly sensitive to the variable wavelengths;
2. The receiver should be the least contributor to the contamination introduced (refer to Section 2.4.1 for detailed follow-up) introduced in the optical signal;
3. A fast speed of response to handle the desired data rate.

#### 2.2.4 Optical Signal Amplifiers & Regenerators

In physics, *attenuation* is the phenomenon of a gradual loss in power of any kind of flux (light wave, heat wave, etc.), when traversing through a medium [14]. A large number of optical signals with varying velocities traversing through an optical fiber result in pulse broadening. This broadening of the optical pulses may allow pulse overlapping of the neighboring optical signals, which is better known as *distortion* [15].

Large amounts of distortion, noise (errors) and attenuation may be introduced in the optical signals that are traversing through various optical devices, thus degrading the quality of the signal [13]. The degradation in the quality of an optical signal may result in the erroneous interpretation of the optical signal by the receiver in the destination. When a source transmits an optical signal using an optical transmitter, the power level of the optical signal may be lowered due to some unavoidable attenuation. To deal with such problems the receiver needs to detect an optical signal reliably with a low bit error rate (errors introduced when the optical signal is propagating the optical carrier)[16]. One of the approaches to improve the power lost by an optical signal during the propagation through an optical fiber is to use an optical amplifier, which essentially amplifies the signal, to make sure that the signal detected may be reliable at its destination.

Traditionally, when there is a need to improve the quality of an optical signal, it needs to be converted into an electronic signal; the electronic signal then is re-timed, reshaped and (re)amplified (a process known as 3R) [16]. This whole process is called 3R regeneration. The optical signal has to go through three major functional blocks represented in Figure 2.8 [16]. As shown in the figure, a distorted signal is received by an optical receiver, where it gets converted into a corresponding electronic signal [16]. The (weak) electronic signal is then given as an input to the amplifier where the quality of the signal is recovered at three different domains, namely – time, error recovery and pulse shaping (not shown) [16]. After the recovery phase the signal is converted back to an optical signal, and passed on to the transmitter for further communication [16]. Some drawbacks of using the Optical Electronic Optical (O-E-O) regenerator has been described in many literature till now [17], and can be summarized as follows:

1. Too many regenerators along the path (the route that the optical signal is routed through) of the optical signal leads to excessive O-E-O, hence removing the flexibility of better maintenance of an optical network.
2. As the number of regenerators in the network increases a network administrator has the added burden of checking the transmission interruption due to the high probability of regenerator failure.
3. The mounting number of regenerators in the network also collectively increases material cost leading to high provisioning and maintenance cost.

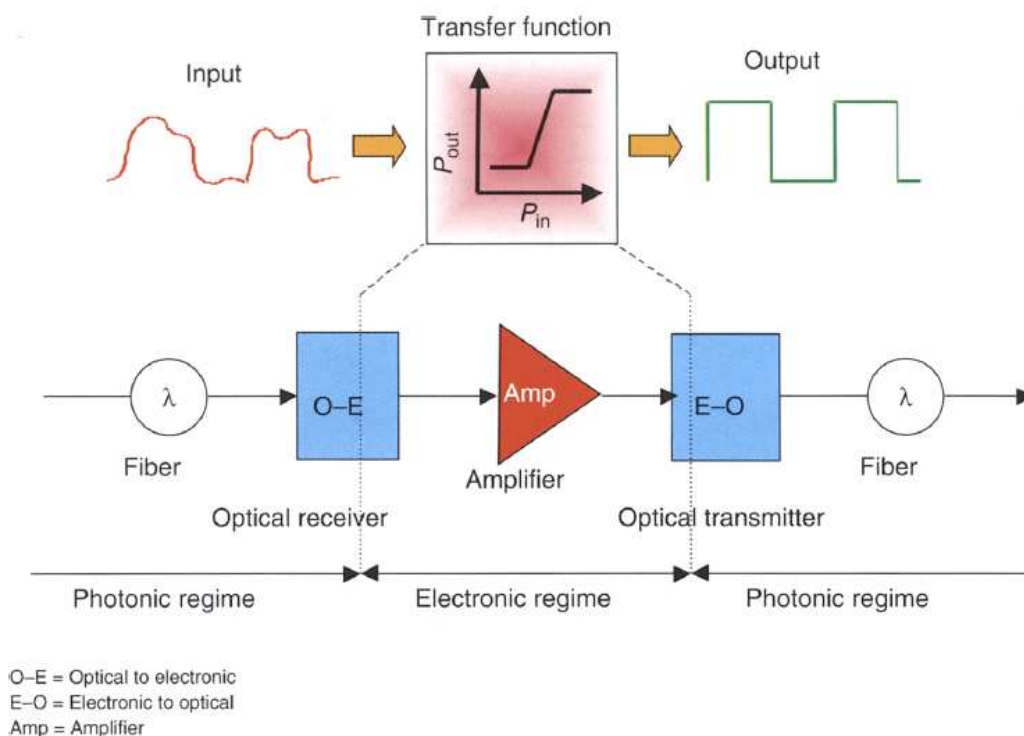


Figure 2.8: The model of a regenerator and the three major functions, optical receiver, electronic amplifier, and optical transmitter

Recently, many researchers have suggested the use Semiconductor Optical Amplifier (SOA), optical fiber amplifiers (OFA), and Raman amplifiers instead of O-E-O regenerators, as these amplifiers can amplify the weak optical signals without an O-E-O conversion (e.g. [16], [17], [18]).

## 2.3 Fundamentals of Optical Networks

### 2.3.1 Wavelength Division Multiplexing & WDM Network

The technology that allows the single optical fiber to act as a carrier for multiple optical signals is called *Wavelength Division Multiplexing*. Sending multiple optical signals over an optical fiber is preferable over sending just one optical signal as adding optical signals at different carrier wavelengths effectively increase the bandwidth capacity of a fiber. This technology also removes the need to install additional fibers to increase the capacity of the network. As indicated before, optical fibers (at the optical domain) have a capacity of sending data at a very high rate, but due to limitations at the electronic domain (e.g. slow electronic processing) [4], it has been found that the electronic domain is unable to use the full bandwidth provided by the optical domain [19]. Due to the gradual incorporation of new technologies in optical communication technology, it is now possible to have a high density of wavelengths in the same fiber. This approach of using

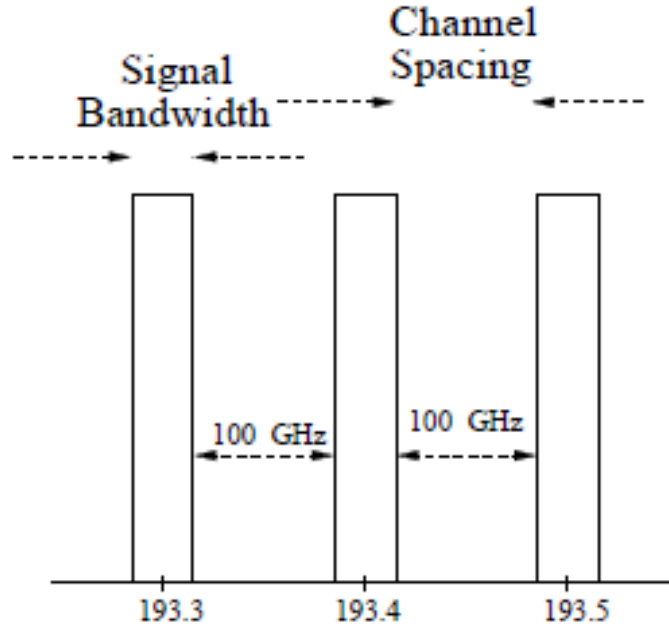


Figure 2.9: Channel spacing

high density optical signals in an optical fiber is called *dense wavelength division multiplexing* (DWDM). On the contrary, for the system using a lower density of wavelength through a single optical fiber, a term *coarse WDM* (CWDM) is used. DWDM makes the most of a large aggregate bandwidth in a single fiber by using some of the technologies available in optical networking. This includes the ability of launching and multiplexing many wavelengths in one fiber, switching wavelengths optically, de-multiplexing and reading each wavelength separately.

The optical channel data rate is denoted by the “OC- $n$ ” notation [4]. Rate OC-1 (= 51.84 Mbps) is considered to be the base rate. Hence, OC- $n$  amounts to  $n \times 51.84$  Mbps. WDM supports transporting a bandwidth equivalent of several OC-192 (or OC-768) signals by carrying each signal on distinct wavelengths in the same fiber. In a single mode fiber (1200-1600 nm), around 1000 wavelength channels separated by 50 GHz may be used. At 40 Gbps per wavelength, a total aggregate bandwidth of 40 Tbps per fiber may be achieved. Commercially, systems with 16, 40, 80 and 128 channels (wavelengths) per fiber have been made available. The 40 channels are said to have a channel spacing of 100 GHz, one with 80 channels are considered to have a channel spacing of 50 GHz. Figure 2.9 [4] further illustrates the concept of channel spacing. As shown in the figure, a channel separation represents the distance between each channel passing through the optical fiber. This separation is required to avoid interference between the channels. For detailed explanation on the figure refer to [20].

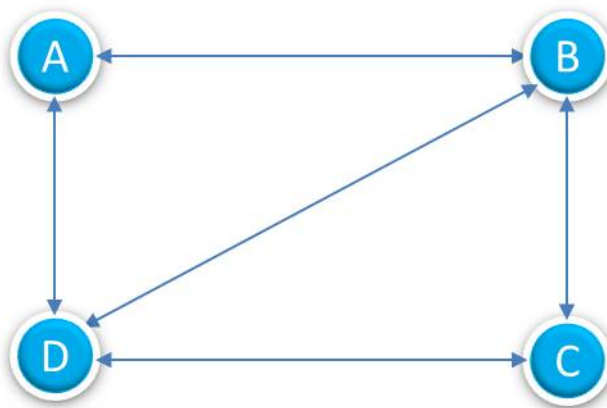


Figure 2.10: A graph representing the network of four networking device communicating

### 2.3.2 Optical Network Representation

A network of computing devices is generally represented by a graph, where the vertices of each graph may represent a computer/router/regenerator or any networking devices that want to communicate with each other through a media. The media/carrier/cable is represented by the edges of the graph. Since the concepts of graph theory can be applied to a computer network, using a graph to represent a network of computers is considered to be a convenient representation [21]. A graph can be either a directed or an undirected graph. In Figure 2.10, each filled circle represents the networking devices that communicate through the edges that connect them. The figure exhibits a directed graph. If there is an edge from a node  $x$  to node  $y$ , then there is also an edge from node  $y$  to node  $x$ .

In optical networks, a graph representing a network of nodes/switching elements, connected by *physical links* (each representing an optical fiber) is called a *physical topology*. If a graph represents a physical topology, then the vertices of the graph will represent various nodes that are connected to the network through an optical router. Figure 2.11 (from [4]) gives the graphical representation of a physical topology where the end nodes are represented by circles and the routers are in the shape of a square. All the routers and end nodes are connected through bi-directional edges representing the optical fiber. In the figure, two end nodes  $E_1$  and  $E_3$  want to establish a communication. With the help of a routing algorithm a path is determined (say)  $E_1 \rightarrow R_1 \rightarrow R_2 \rightarrow R_3 \rightarrow E_3$  (further discussions on the routing algorithm can be found in Section 2.3.3). The connection (along with other connections) between  $E_1$  and  $E_3$ , at an optical domain, is shown in Figure 2.12. This type of optical connection from one end node to another, carrying an encoded optical signal, is called a *lightpath*. Such lightpaths start from and end



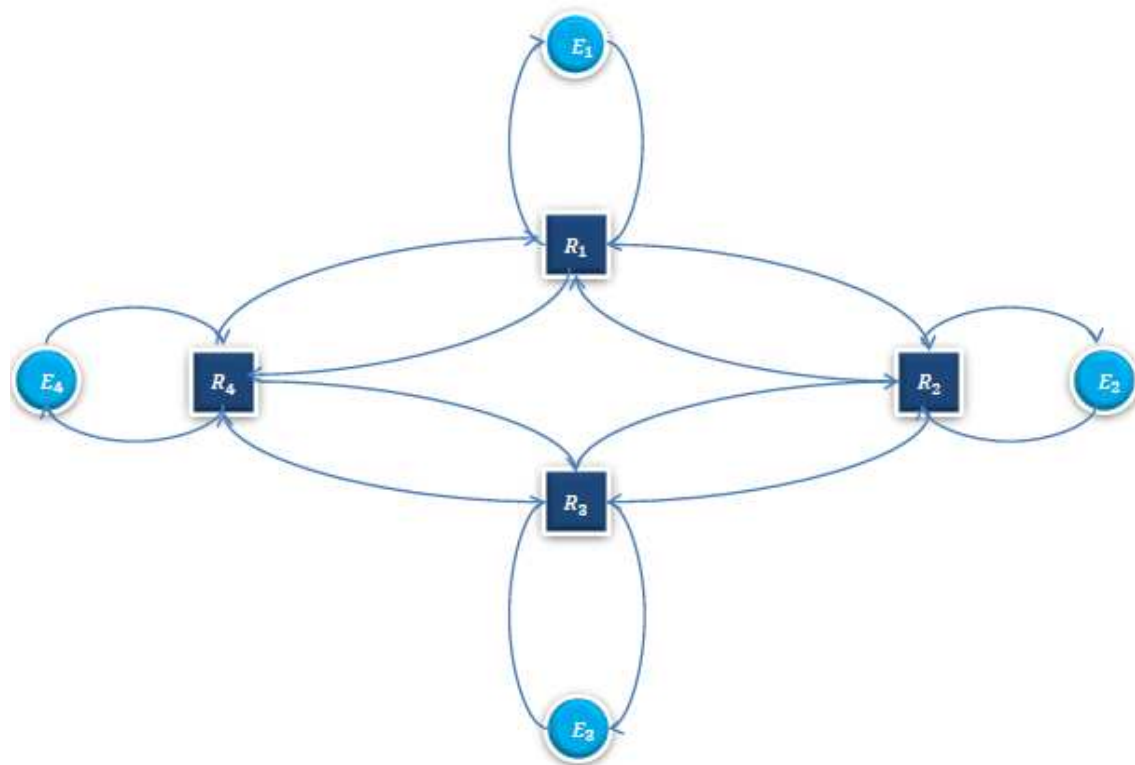


Figure 2.11: The physical topology of a typical WDM network with four end nodes  $E_1, \dots, E_4$  and four routers  $R_1, \dots, R_4$

at an end node. In this case  $E_1$  is the source and traverses through numerous fibers and routers ( $R_1, R_2, R_3$ ) to finally end at an end node, in this case  $E_3$ . The graph represented in Figure 2.12 [4] is called a *virtual topology*, where the edges represent the lightpaths established between the nodes in the corresponding physical topology.

### 2.3.3 Routing and Wavelength Assignment

Before starting the general discussion on various routing algorithms used for routing the data in an optical network, some important concepts are explained. The request for establishing a lightpath generally arrives at a node with two pieces of information in it – the *source* node and the *destination* node. The set of such pairs of source and destination nodes that want to establish lightpaths for data communication is known as the *demand set*. Figure 2.13 [17] shows the classification of different types of demand sets. As shown in the figure, a demand set can be classified into two main categories. In the case of *permanent lightpath demands* (PLDs), the lightpath demands (source and destination pairs) continue the flow of data for a relatively long period of time; hence the lightpath connections are termed *permanent* or *static* [17]. In contrast, for *dynamic lightpath*



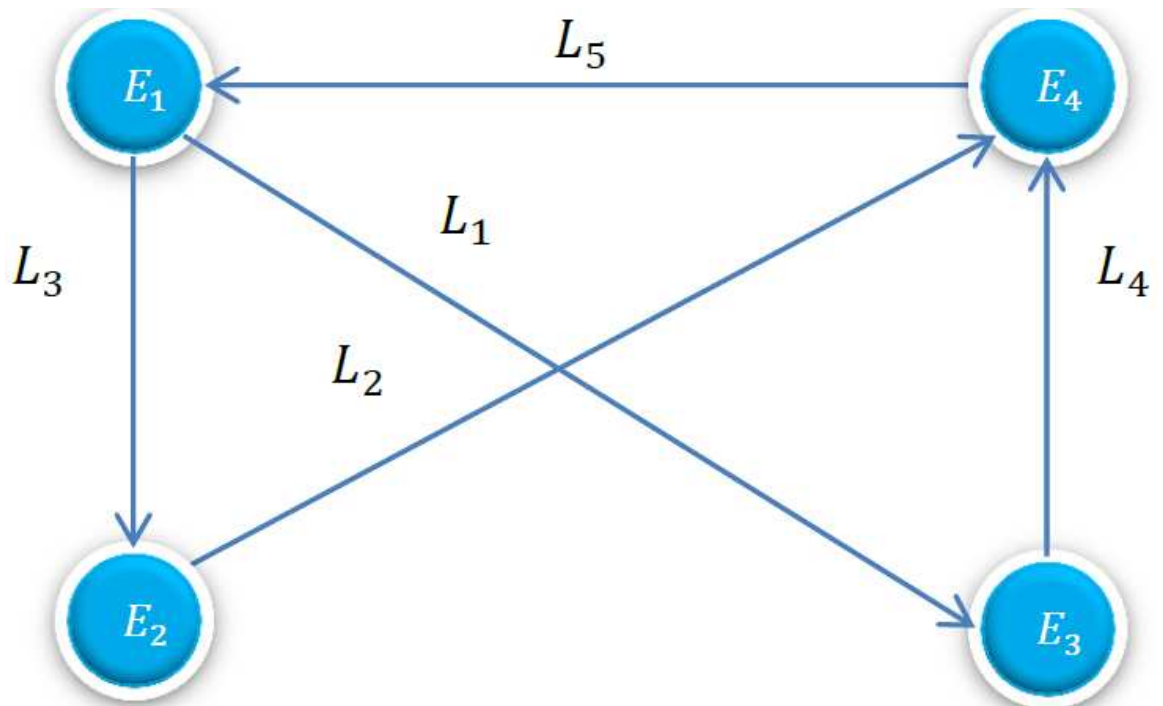


Figure 2.12: Logical topology corresponding to the lightpaths shown in Figure 2.10

*demands* (DLDs), the data communication between a source and a destination continues for a very short period of time [17]. The DLDs are further sub-divided into two more categories – *scheduled lightpath demands* (SLDs) and *ad hoc lightpath demands* (ALDs) [17]. SLDs are the demand sets where the activation time and the lifetime of the demands are known. ALDs don't provide any information about their arrival time or lifetime beforehand.

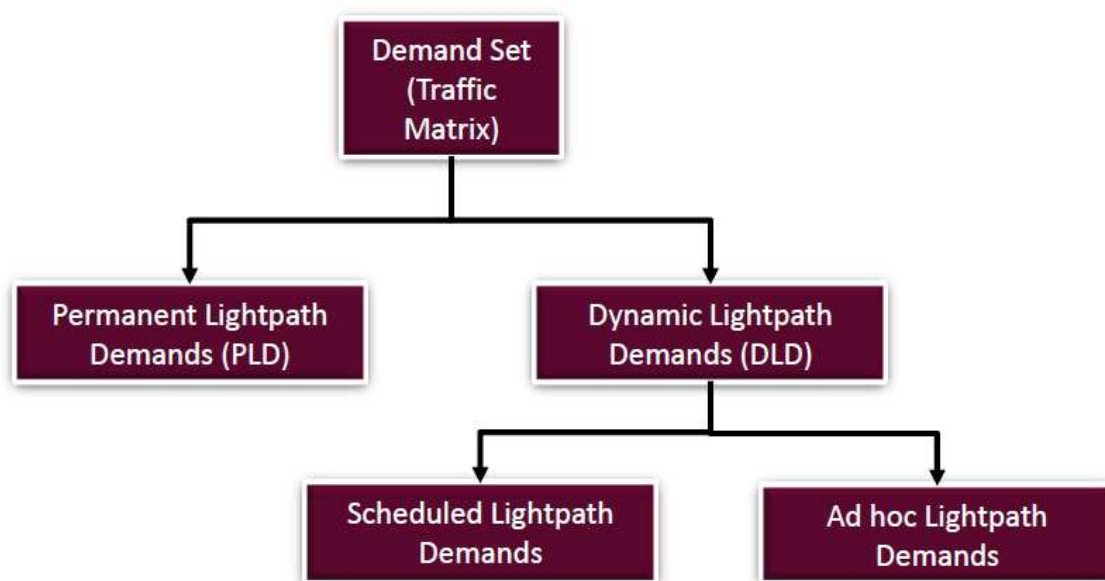


Figure 2.13: Demand set categories

As discussed previously, the data in a wavelength-routed WDM networks is communicated using all-optical connections known as lightpaths. From the discussions so far, it is quite apparent that the lightpaths traverse through various fibers and other optical devices to finally reach the destination without undergoing any O-E-O. A lightpath is routed over the network in this manner and is assigned an available wavelength on the fiber. The route selection is accomplished by employing an algorithm, which finds a path to the destination (if available); the path may or may not be selected for the lightpath depending upon the availability of a wavelength which is not used on any fiber along the path. If such a path exists and the path can be assigned a wavelength, then the demand is said to be *established*, otherwise, it is termed as a *blocked* demand. The process of routing a lightpath for a selected wavelength is called *routing and wavelength assignment* (RWA) [22]. When traversing through fiber links, if two lightpaths share the same fiber, then they cannot employ the same wavelength. In the absence of wavelength conversion, the lightpath is required to use the same wavelength all along the path, this constraint is known as the *wavelength continuity constraint*. The current technology requirement of an RWA algorithm is that it should follow the wavelength continuity constraint [22].

According to the type of traffic demands to be routed on an optical network, there are two types of RWA algorithms – the *static RWA* and the *dynamic RWA*. If the traffic demands are permanent (in other words if we are dealing with PLDs), then the algorithm needs to be designed to solve a static RWA problem.

The objective of a solution to such a problem is to establish as many lightpaths as possible, using the network resources to the most optimum extent. Routing and Wavelength Assignment can be considered together to get an optimized solution using some *Mixed Integer Problem* (MIP) [23], with the objective of minimizing the average number of hops or minimizing the mean latency in the delivery of a packet using the lightpath. Since the problem has been proven to be NP-complete [24], researchers have proposed many heuristic algorithm solutions to solve the problem [25]. Heuristic algorithms tend to give solutions that are very close to the optimum solutions but have the ability to solve large problems fairly quickly. The general approach to an RWA problem is to divide it into two sub-problems and then solve them individually [23]. With the assumption that there is no facility for wavelength conversion, the two sub-problems are as follows [22]:

1. **Route Assignment sub-problem:** This part of the solution can use any well-known routing algorithm to find a route from the source to destination for a given network (specified in the form of a graph).
2. **Wavelength Assignment sub-problem:** This part of the sub-problem deals with wavelength assignment to the routes evaluated by the *route assignment* sub-problem. While assigning the wavelengths, the approach must consider the wavelength continuity constraint. *Graph coloring algorithm* and *first fit* are some of the widely accepted solutions to the wavelength assignment sub-problem [22].

A dynamic RWA is used for dynamic traffic scenario, where the lightpath demands must be handled as and when a request is received. A network state is generally specified by the physical paths and the wavelengths assigned to already established (active) lightpaths. Since the state of the network at a moment is not known in advance, the incoming request may or may not be established [24]. Other reasons for rejecting the lightpath will be discussed in Section 2.5. A request can be rejected due to the non-availability of resources (e.g. lack of available wavelength). Since this thesis will focus on static routing and wavelength assignment, dynamic routing and wavelength assignment will not be discussed further. A comprehensive discussion on dynamic RWA is available in [22], [23], [24] and [25].

### 2.3.4 Types of Optical Networks

Depending on the extent of O-E-O conversion (regeneration) done in it, the optical network can be classified into three categories:

1. **Opaque Optical Networks:** In this type of optical networks, every node is capable of performing optical signal regeneration. This type of network is also termed as *all-electronic switching network* as the optical signals undergo O-E-O conversion at every node [26]. There are many benefits of such a network, since any degradations in the quality of the optical signals can be easily handled, as the signals are constantly getting regenerated at every node they visit during the propagation towards a destination [27]. However, there are a few drawbacks of using opaque optical network, as these types of networks tend to carry out O-E-O conversion at every node, which ends up being an expensive approach [27]. Also the latency of an opaque optical network is a problem since such networks do not respond to data rate up-gradation due to the electronic domain's lack of transparency to the data rate of optical channels [26].
  
2. **Transparent Optical Networks:** The bottleneck caused by the electronic domain can be improved by employing *transparent optical networks* (TONs). TONs allow the signal to remain in an optical form during its propagation towards a destination [28]. The node architecture is simple in the case of the TONs, as the need for an electronic switching fabric is eliminated [26]. TONs use all-optical switches that can switch the light streams without converting them into electrical signals [29]. The use of such switches increases the speed and lowers the costs [29]. It was realized by the authors of [28] that the routing algorithms need to keep track of very limited information in the packet header, hence giving more control over the routing algorithms. The absence of O-E-O conversion in TONs make it difficult to handle the degradation of signal quality. Since there is no regeneration involved, the optical signal quality may degrade so that errors may occur when the receiver at the destination decodes the incoming signal. This degradation is incurred due to the presence of many physical layer impairments [30] (for further details refer to Section 2.4.2).
  
3. **Translucent Optical Networks:** The two types of optical networks discussed so far either regenerates any optical signal at every node (OONs) or there is no regeneration at all (TONs). In [30], the authors propose the use of a flexible optical network where the regeneration of an optical signal is performed only whenever the necessity arises. A translucent network tries to combine the advantages of the previously discussed networks to achieve a better balance between the cost and the network performance. A term called *optical reach* denotes the distance an optical signal can traverse without the need of optical regeneration. Only when the optical signal reaches

its maximal optical reach, the O-E-O signal regeneration is performed [26]. In the network shown in Figure 2.14, the edges are labeled with distances between each node pairs connected by fiber. Let the optical reach for the network be 100. To maintain this discussion as brief as possible, let one of the source-destination pair from a demand set be  $E_1$  and  $E_6$ , where  $E_1$  is the source and  $E_6$  is the destination. The shortest path from  $E_1$  to  $E_6$  is  $E_1 \rightarrow E_4 \rightarrow E_6$  with a total distance of 125. Since the optical reach is less than the path length, it is quite apparent that if there is no regeneration, an optical signal cannot reach the destination with an acceptable quality. The best place to allow a regeneration would be at node  $E_4$ . Hence, we have an augmented graph that looks similar to the one shown in Figure 2.15, where the  $R_1$  represents the regenerator at node  $E_4$ . Researchers have proposed many algorithms of determining the optimal placement of regenerators [17]. The regenerator placement problem in a translucent optical network is essentially the problem of determining the location of regenerators within the network so that a minimum number of regenerator are used, so that the optical signal from any node reaches its destination with an acceptable quality but with minimum O-E-O conversion [30]. The translucent optical networks can have two variants. The first type includes *sparsely distributed opaque switch* nodes, in which only a few switches have regeneration capability and all the other are all-optical [26]. In the second type of translucent optical network, all nodes in the network are *translucent switches*, which when required can regenerate the signal by performing the O-E-O conversion; otherwise these nodes will bypass the signal optically [26].

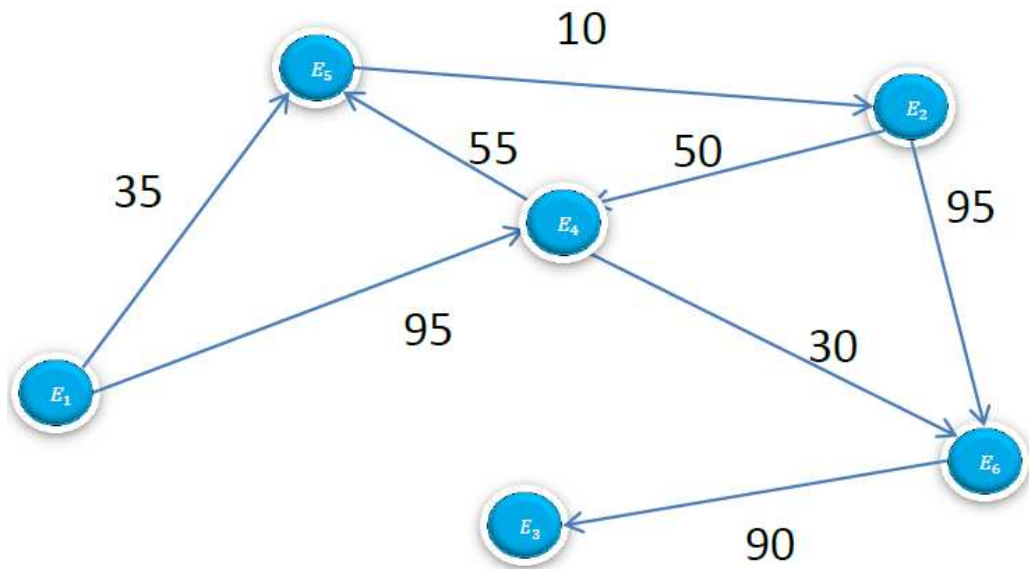


Figure 2.14: A simple network

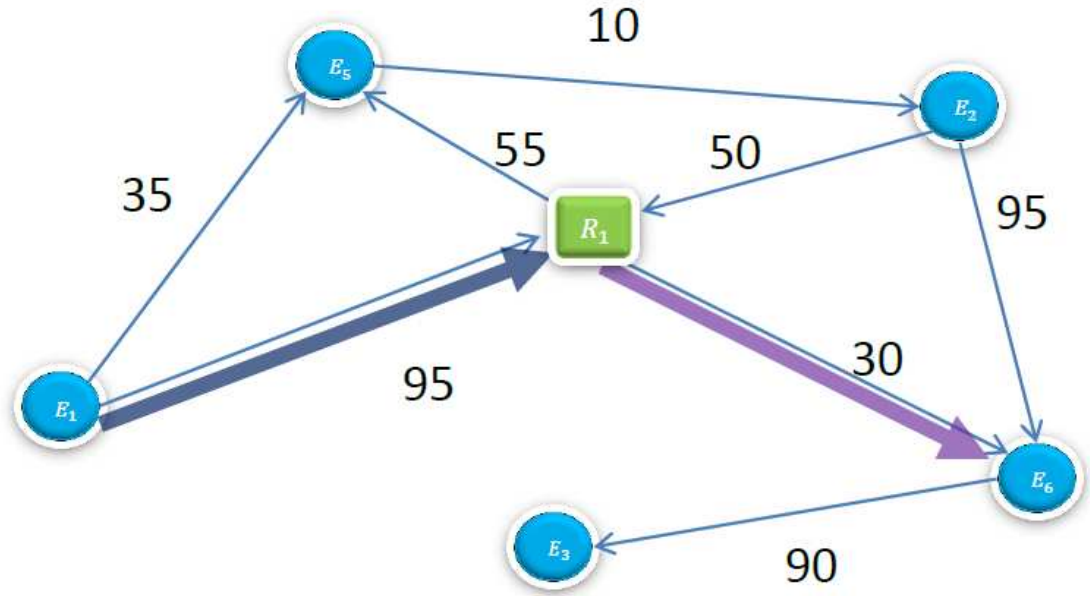


Figure 2.15: At Node  $E_4$  a Regenerator  $R_1$  is placed, as the distance of the path from  $E_1$  to  $E_3$  is greater than optical reach

### 2.3.5 Physical Layer Impairment

An optical signal traversing through an optical fiber has to share the optical fiber with numerous other optical signals if the DWDM technology is used [30]. Due to this situation within the optical medium and the non-availability of the O-E-O conversions at the intermediate nodes, the optical signal quality may degrade to a level that an optical receiver may interpret the incoming signals incorrectly. The physical phenomena that cause the degradation in the quality of signal are called *physical layer impairments* (PLI). There are numerous factors which may act as the source of PLI. Some of them are – manufacturing defects in cables and other optical devices, the deterioration of signal power as the propagation distance increases, use of amplifiers, the presence of other optical channels (wavelengths) within the same fiber.

To measure the extent of the degradation of the signal quality, two different metrics are generally considered, namely, *Optical Signal-to-Noise Ratio* (OSNR) and *Bit Error Rate* (BER). The intensity of noise is measured in *decibels* (dB) when OSNR is considered as the metric [16]. The misalignment, jitter and other disturbances caused by the noise introduced by PLIs can result in erroneous bits in the optical signals, whose rate is given by BER [16]. For a given OSNR value in dB, the BER can be calculated by the following formula:

$$\log_{10}(BER) = 10.7 - 1.45(OSNR)$$

The authors of [17] quote the works that have successfully integrated BER with a routing algorithm to consider the *quality-factor* (Q-factor) of the optical signal. A Q-factor is also a measure for the *quality of transmission* (QoT) of optical signals. The following Section states the different classification of PLIs along with a few examples.

### 2.3.6 Classification of Physical Impairment

Physical layer impairments (PLIs) can be classified into two categories, namely, *linear impairments* and *non-linear impairments*. The linear impairment is not affected by the signal power, and affects each wavelength individually. The non-linear impairments affect each optical channel individually as well as cause distortions and interference among the channels. The following paragraphs will discuss the two types of physical layer impairments in detail.

#### Linear Impairments

Following are a few of the important linear impairments –

1. ***Power Losses***: power loss is generally defined as the optical loss that is accumulated from the source to the destination as the optical signal traverses the optical fibers. These losses are due to fiber losses and bend losses. Fiber losses are due to some physical phenomena like attenuation, absorptions, reflections, refractions etc. [30].
2. ***Chromatic Dispersion (CD)***: The degradation of optical signal caused by the differences in the velocities of different spectral components is called *chromatic dispersion* [30]. Chromatic dispersion leads to pulse broadening, which in turn affects the performance of the receiver by introducing two factors – reducing the energy of the pulse and spreading the energy of the pulse beyond the allocated slot [17]. In other words, CD allows an optical pulse to broaden such that it spreads into the slots of other pulses [30].
3. ***Polarization Mode Dispersion (PMD)***: There are very high possibilities that at some point in the fiber-span, the fiber is not precisely circular, may contain impurities or is under the influence of some physical stress. These types of non-ideal conditions simply creates obstacles along the path of optical signal resulting in the optical signal traveling at different group velocities – a phenomenon known as *polarization mode dispersion*. Polarization Dependent Loss (PLD) is another impairment that belongs to the same family of PMD. PLDs can cause optical power variation, distortion in waveform and fading of signal-to-noise ratio [17].



4. ***Amplifier Spontaneous Emission Noise (ASE)***: An optical signal may be contaminated by noise during the process of optical amplification; this kind of noise is known as *amplifier spontaneous emission (ASE) noise*. The occurrence of such noise is promoted by the presence of a series of amplifiers along the path of an optical signal [31]. The degradation due to such impairments are directly associated with OSNR.
5. ***Crosstalk (XT)***: Partial separation of optical signals (wavelengths/channels) is the effect of signal power leakage from one WDM channel to another WDM channel. *Linear crosstalk* is caused by partial separation of channels by optical components like OADMs, OXCs, multiplexers/de-multiplexers and optical switches.

### **Non-linear Impairments**

Often, in cases when the optical signal power levels are low, the refractive index and attenuation of a fiber is independent of the optical signal power [32]. However, most DWDM systems tend to employ high power level optical signals, therefore introducing non-linear impairments into the network [32]. Some of the important non-linear impairments are as follows –

1. ***Self-Phase Modulation (SPM)***: The non-linear phase modulation of an optical signal pulse caused by its own intensity is called *self-phase modulation* [31].
2. ***Cross-Phase Modulation (XPM)***: There may be multiple pulses propagating through an optical fiber at varying wavelengths. Due to the presence of non-linear effects, each wavelength results in a change of refractive index of the optical fiber [33]. The change in the refractive index of the fiber depends on the power carried by the wavelength of an optical signal. If the change in refractive index brought by an optical signal affects the propagation of other optical signals, then the effect is known as *cross-phase modulation* [33].
3. ***Four-Wave Mixing***: This type of impairment is caused by the non-linear interaction that occurs in the presence of multiple channels in the fiber, resulting in creation of signals at new frequencies [33].



## 2.4 $k$ -Shortest Path Algorithms

### 2.4.1 Importance of $k$ -Shortest Path Algorithms

One of the major components of this work is the implementation of the  $k$ -shortest path ( $k$ -SP) algorithm. For a given graph  $G(V, E)$ , with  $|V|$  vertices and  $|E|$  edges, a  $k$ -SP algorithm can find the first  $k$  shortest paths between any two vertices. A  $k$ -SP algorithm plays an important role in the current literature. When solving an RWA problem, a route may be rejected due to unavailability of wavelength. In the case of IA-RWA, a route may be rejected due to some additional constraints, e.g. a route may have exceeded the optical reach or the QoT value has exceeded the threshold limit. If only the shortest path is considered, while finding the path in RWA, then the demand will be blocked just because there was no attempt made to route the lightpath through a different route. When  $k$ -SP is used instead of a regular shortest path algorithm, the heuristic has additional options of trying out different paths if a path is rejected.

A huge amount of work exists to solve the problem of  $k$ -SP. Due to the lack of space, a limited numbers of relevant papers are listed here. Some of the widely cited papers that generalize the  $k$ -SP algorithm are written by the authors of [34] and [35]. A common element found in all the implementations of a  $k$ -SP algorithm is that, after searching the first shortest path, the graph is modified in such a way that the next path is the only shortest path available after the first shortest path and the previously found shortest path is not available in the modified graph. A vast number of implementations are available, based on these two papers. In [35], the authors efficiently finds first  $k$  shortest paths, but allows loops (cycles) in the paths. If a loop in the path indicates that a node can be visited more than once in a path. A comprehensive implementation of the algorithm in [35] is provided by [36]. In the case of [34], the algorithm does not allow any loop in the paths that it searches for.

### 2.4.2 Yen's Algorithm

Yen's Algorithm [34] computes  $k$  shortest paths ( $k$ -SP) between a source and a destination in a given network graph. This Section gives the details of Yen's algorithm.

#### Notation used –

$G(N, E)$  Graph with the set of nodes  $N$  and the set of edges  $E$ .

$(s, d)$  The source-destination pair, where  $s, d \in N$ .

- $K$  The number of shortest paths that needs to be found between  $(s, d)$ .
- $d_{ij}$  The *distance* between the nodes  $i$  and  $j$ , where  $i, j \in N$  and  $d_{ij} > 0$  if there is a direct edge from node  $i$  to node  $j$ , otherwise  $d_{ij} = \infty$ .
- $P^k$  The  $k^{th}$  shortest path, represented by  $P^k = (s) \rightarrow (2^k) \rightarrow (3^k) \rightarrow \dots (p_k^k) \rightarrow (d)$ , where  $1 \leq k \leq K$ .  $(2^k), (3^k), \dots, (p_k^k)$  are the 2nd, 3rd,  $\dots$ ,  $p_k$ th nodes in the path  $P^k$ .
- $P_i^k$  The shortest of the paths that coincides with  $P^{k-1}$  from node  $(s)$  to the node  $(i)$ , where  $i = s, 2, \dots, p_k$ , and then deviates from its course to choose any of the  $(i + 1)$ st nodes of those  $P^j$ , such that  $j = s, 2, \dots, k - 1$ . Same as the path  $P^{k-1}$ ,  $P^j$  is a set of paths that have the same path from  $(s)$  to the  $i$ th node, and finally reaches  $(d)$ , without including any of the node already visited in the first part of the path. Hence the node  $(i)$  is called the *point of deviation* (PoD).
- $R_i^k$  The *root* for the path  $P_i^k$ . Hence,  $R_i^k \in P^{k-1}$ .
- $S_i^k$  The *spur* for the path  $P_i^k$ , which essentially is the tail part of the path  $P_i^k$  with just one node (namely  $(i^k)$ ) coinciding with the path  $P^{k-1}$ .
- List A* The list of  $k$ -shortest paths
- List B* The list of candidate  $(k + 1)$ st shortest paths.

**Yen's Algorithm** The algorithm goes through two stages. In the first stage the algorithm finds the first shortest path for the  $(s, d)$  pair. The second stage consists of a number of iterations. In each iteration the next shortest path is found.

In the paper [34], the authors have suggested many algorithms including [37] and [38]. A path is considered to be a *loopless path* when none of the participating nodes in the path are visited more than once. Yen's algorithm does not consider any negative loops in the network and hence it is assumed that there exists at least one shortest path between the source and the destination pair, that does not contain any loop.

The detailed explanation of the algorithm is as follows:

**Iteration 1.** To determine  $P^1$

In this iteration any shortest path algorithm (e.g. [38]) is used to get the first shortest path between the pair  $(s, d)$ , in the graph  $G(N, E)$ . Following the notation, this path is denoted as  $P^1$ .  $P^1$  is stored in the *List A*, as it is one of the paths that needs to be found.

**Iteration**  $k$  ( $k = 2, 3, \dots, K$ ). To determine  $P^k$

Before finding the  $k$ th path in the graph  $G$ , this algorithm requires all the previous shortest paths (i.e.  $P^1, P^2, \dots, P^{k-1}$ ) to be evaluated. The path  $P^k$  is determined as follows –

1.  $\forall i = 1, 2, \dots, p_k$  perform the following:

a) Determine if a sub-path consisting of the first  $i$  nodes of  $P^{k-1}$ , sequentially coincide with the sub-path consisting of the first  $i$  nodes of  $P^j$ , such that  $j = s, 2, \dots, k-1$ . If there is such a sub-path, mark the distance between the two nodes as infinity (i.e.  $d_{i(i+1)} = \infty$ ), otherwise make no changes in the graph, and directly proceed to Step(b).

The distance  $d_{i(i+1)}$  is set to  $\infty$  only for the current ( $k$ th) iteration. Before the start of the ( $k+1$ )th iteration, all the changes made must be restored.

b) The shortest path algorithm is applied to the node pair  $(i, d)$ , allowing the algorithm to pass through the nodes that are not already visited. The sub-path from  $(s)$  to  $(i)$  is called  $R_i^k$ , the root of  $P_i^k$ . This part of the iteration generates the spur  $S_i^k$ .

There may exist more than one such spur that have the same length. In such cases, any arbitrary one of them is selected.

c) The path  $P_i^k$  is then found by joining the two sub-paths – the  $R_i^k$  and the  $S_i^k$ .

d) This path is considered to be one of the candidate paths, and hence stored in *List B*.

The sub-steps from (a) to (b) are repeated for  $k-1$  number of  $P_i^k$ 's.

2. Remove the path with minimum length from *List B* and store it in *List A*, hence marking it as the next shortest path.

3. Repeat steps 1 and 2 until the number of shortest path in *List A* has reached the value  $K$ .

## 2.5 Relevant Work Done in IA-RWA

The authors of [22] present a comprehensive survey on the work done in classic RWA. These works have a common assumption that the transmission mediums do not incur any impairment in the optical network. However, it's been observed by authors of [17] that the actual performance of the optical network may be

unacceptable for some of the lightpaths if the impairments are not considered. Due to the above mentioned concerns, inclusion of PLIs as additional constraints while solving RWA has lately received a lot of attention from the research community. Algorithms that consider PLIs as constraints for the RWA are, in general, termed as *impairment aware routing and wavelength assignment* (IA-RWA). As mentioned earlier (refer to Section 2.4), an RWA is an NP-complete problem. Considering PLI constraints while solving the RWA problem introduces further complications [17]. This Section starts with an introduction to various approaches to route data (in other words perform RWA) while considering physical layer impairments and finally discusses various works done in this area.

### 2.5.1 Approaches to PLI-RWA

There are many ways to classify the literature on IA-RWA. Here are a few classification criteria –

1. Authors of [17] have successfully classified the work done in terms of “*performance evaluation technique*”. The performance can be evaluated according to the type of impairments considered, which can be either linear impairments or non-linear impairments.
2. Due to the strong dependency of PLIs on the bit error rate, the modulation formats and the type of amplifiers used, some work can also be classified based on the type of model used to measure the impairments [17].

Figure 2.16 taken from [17], shows the three general approaches to PLI-RWA (or IA-RWA). According to [17], there can be three main classes, based on when the impairments are considered –

1. In the first case, the quality of the lightpath may be considered after routes and wavelengths are determined;
2. In the second case, the routing and/or wavelength assignment decisions are made considering the PLI values;
3. Finally, in the third case, the PLI values are considered in route and/or wavelength assignment decisions and finally also verify the quality of the candidate lightpaths.

### 2.5.2 Previous Work

As mentioned earlier, RWA are classified into two categories, based on the type of demand set, namely – *static (offline)* RWA and *dynamic (online)* RWA. Similarly,

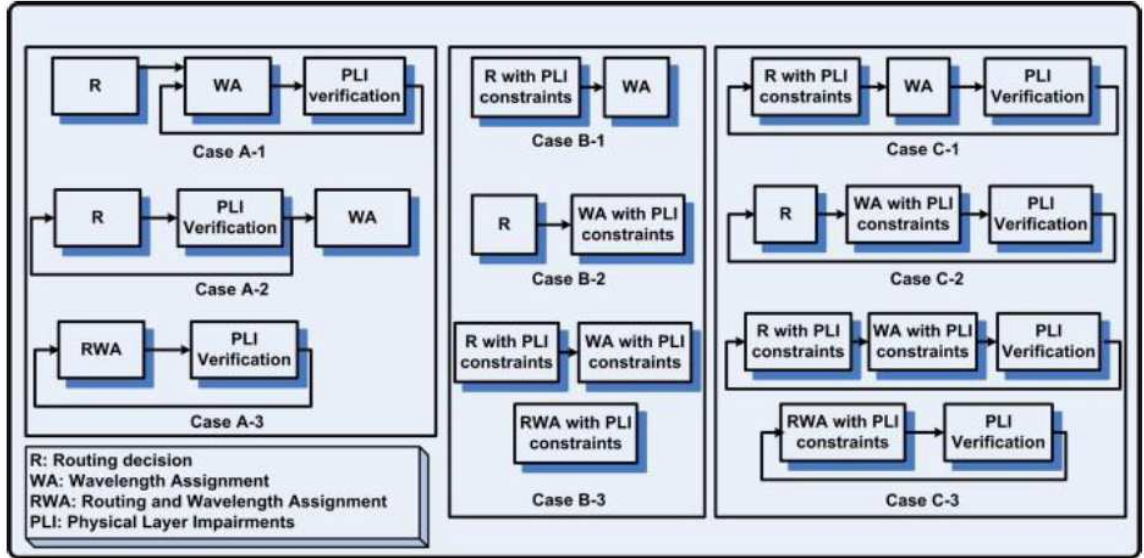


Figure 2.16: Various PLI-RWA approaches

the IA-RWA can also be categorically termed as static IA-RWA and dynamic IA-RWA, based on the type of demand set the solution is dealing with. Additionally, IA-RWA can also be categorized according to the type of optical network the RWA algorithm is working on. If the network is a transparent optical network then the IA-RWA can be either offline IA-RWA for all-optical network or dynamic IA-RWA for all-optical network. If the network under investigation is a translucent optical network, then there is an additional problem of regenerator placement. These types of IA-RWA are hence called translucent offline/online IA-RWAs. Before analyzing the related work done in the area of interest, it is worthwhile to reiterate the problem that this thesis is trying to solve. Although there will be frequent references to other types of IA-RWAs, this review primarily concentrates on *offline (static) impairment aware routing and wavelength assignment for transparent optical networks*. The main goal of the thesis is to introduce the concept of “generic” test-bed for new heuristics for static IA-RWA.

In general, an IA-RWA can be solved using either of the following approaches:

1. By developing a heuristic or meta-heuristic algorithm, an IA-RWA problem can be solved while achieving a solution close to an optimal solution [17].
2. By adopting a combinatorial approach, the IA-RWA can be solved to get an optimal solution.

Due to the NP-completeness of the problem, generally the heuristic/meta-heuristic approaches are preferred [17].

The papers [39], [40] and [41] describe the work done in the literature that have used LP-relaxation to solve the static IA-RWA problem in transparent networks. Due to the NP-completeness of the IA-RWA problem, the use of LP-relaxation (combinatorial) approaches is quite restrictive. In case of optical networks, the size of the search area is defined by the number of nodes and number of links connecting the nodes in the network. The time taken to solve an IA-RWA problem thus increases rapidly with the increase in size of the network. A network size of 10 nodes and 16 bidirectional links is considered in [39] and a network size of 14 nodes and 46 directed links (Deutsche Telekom network) is used in the simulation by the authors of [40] and [41]. Due to the small size of the networks considered in the work listed above, the authors were able to achieve an optimum solution in reasonable amounts of time ([40] and [41] both took around 3 hours). Some algorithms proposed in the literature use Integer Linear Programming (ILP) as a part of its heuristic, e.g. in [42], a “Core Algorithm” is the part of the heuristic that forms a major part of “Global Search Scheme”. This algorithm uses an ILP to minimum number of wavelengths used in the network.

As mentioned earlier (Section 2.3.3), an RWA can be solved by dividing the problem into two sub-problems – the routing sub-problem and the wavelength assignment sub-problem. The routing sub-problems are generally based on the shortest path routing algorithm (e.g. Dijkstra algorithm) [17]. The heuristic algorithms can consider two approaches, when dealing with the routing sub-problem, namely, *single-path routing algorithms* and *multi-path routing algorithms* (also known as *k*-shortest path or *k*-SP). The authors of [43],[44], [45], [46], [47], [48], [49], [44], [50] and [51] have approached the routing problem by considering a single-path algorithm [17]. Among these [47], [48], [49] and [51] solve IA-RWA for *online* (dynamic) traffic demands. The routing sub-problem using some *k*-SP algorithm are discussed in [52], [53], [54], [55], [56], [31] and [57]. Among these [31] and [57] are the works which solve the IA-RWA problem for translucent networks. As noted before (Section 2.4.3), *k*-SP algorithms have an added benefit over single-path algorithms.

There is another class of PLI-RWA (IA-RWA) algorithms that exploit meta-heuristic approaches. Meta-heuristic (hyper-heuristic) approaches are found to be attractive solution as they do not involve complex mathematical equations (or formulations) and, at the same time, it reaches a “near” optimum solution in successive iterations [17]. An ant colony hyper-heuristic is presented by the authors of [58], which solve the static IA-RWA problem for transparent optical networks. The algorithm is based on the way ants construct paths. The paths are constructed by the ants according to the pheromone trail values assigned to the edges [59]. As a path is traversed by the ants more frequently, the path becomes

more preferred over others. With *pheromone evaporation* the *pheromone value* of a path starts deteriorating, hence a path may be rejected if it is not used by the ants any more [59]. Genetic algorithms [60] are another approach to solve multi-objective IA-RWA. One of the major contribution towards IA-RWA that solve the problem by adopting genetic algorithm is found in [61].

The work reported in this thesis concentrates on the building a test-bed which will have the ability to compare different work done in static IA-RWA for transparent optical networks, on a common platform. Different comparisons of performance are presented in the literature. Table 2.1 summarizes the papers based on the approaches adopted to compare the performance the research work under the heading “Objective of the paper”. It is quite apparent from the table, that apart from the work of [52] and [44], all other listed research papers have either compared their performances to [31], [56] or to the classical RWA. A similar trend is followed in this thesis (more detail in chapter 3).

Table 2.1: Different approaches to comparisons

Year	Objective of the Paper	Works Compared To
2004	Authors propose an algorithm that can accept a list of traffic demands, with PLI information, and efficiently solve the static IA-RWA problem for transparent optical networks while performing load balancing. [52]	Simulation results are run for two different modulation formats – LPF and NRZ Two different cases considered – <ol style="list-style-type: none"> <li>1. Without dispersion (CD).</li> <li>2. With CD at each node.</li> </ol>
2007	With the goal of minimizing the overall blocking ratio in transparent optical networks, the authors proposed a novel algorithm – <i>least variance algorithm</i> , which takes both linear and non-linear impairments into account, for a static demand set [44].	The algorithm is compared against itself with few tweaks in parameters. For example, by using different permutations of routing and wavelength assignment approaches, authors found that the algorithm performed the best, when the <i>first-fit</i> approach is employed for wavelength assignment.

2007	The main goal of the authors is to reduce the time complexity of traditional Q-factor formulation in static IA-RWA for transparent optical networks, while maintaining the accuracy of the quality measure. The authors termed their novel algorithms as IAWA (Impairment Aware Wavelength Assignment) and PS (Pre-Specified)-IAWA. The later approach has an extra phase called preprocessing phase [54].	LERP [31]
2009	Paper proposes two algorithms, namely, P-IA-RWA (Parametric-IA-RWA) and SB-IA-RWA(Sigma Bound-IA-RWA), to compare with an ILP-implemented pure-RWA. [40]	<i>Pure-RWA</i>
2009	The objective of the paper is to investigate the impact of PLI while dealing with lightpath provisioning problem. [55]	Traditional RWA
2009	The authors propose POLIO-RWA (pre-ordering least impact offline-RWA). The benchmark for this paper is [31], which coined the concept of pre-ordering the demand set. [56]	RS-RWA (random search), RS-RWA-Q (random search with Q-factor). Both approaches were introduced in [31].



2010	Objective of this paper is to come up with a Reordered Lightpath Establishment (ROLE) heuristic which is compared to POLIO-RWA [56] and a proposed ILP formulation. The ILP formulation considers aggregated effect of PLI and considers “hardened constraints” to guarantee BER. [53]	POLIO-RWA (pre-ordering least impact offline-RWA), Proposed ILP, LERP – lightpaths established without rerouting, LERO, lightpaths established without reordering
2010	The authors implemented three different ILPs – <ol style="list-style-type: none"> <li>1. <b>1st ILP</b> – implemented classical-RWA (or pure-RWA)</li> <li>2. <b>2nd ILP</b> – implemented IA-RWA using an indirect impairment estimation scheme.</li> <li>3. <b>3rd ILP</b> – implemented the IA-RWA, with direct cost estimation. [41]</li> </ol>	ILPs (which considered impairment) were compared against <i>pure</i> -RWA

With 56 citations, [31] is the most widely cited paper in the last five years of work done in this area of research [62]. Authors of [31], present a novel approach to solve static IA-RWA in translucent networks. Two types of heuristic approaches are presented by the authors, namely, sLERP (simple Lightpath Establishment with Regenerator Placement) and LERP (Lightpath Establishment with Regenerator Placement). Both algorithms have a common step of performing random ordering of the demand set and then performing a sequential RWA for each permutation. For each source-destination in a demand set, the algorithms evaluate  $k$  shortest paths. A path from the evaluated  $k$  paths is chosen, such that the path allows the wavelength continuity constraint (Section 2.3.3). If the constraint is not satisfied for the any of the paths, then the source-destination pair is blocked. A permutation with the least number of blocked calls is selected for the next step. The above process is called *random search routing and*

*wavelength assignment* (RS-RWA). In the case of sLERP, after performing the RS-RWA, the heuristic enters the QoT test phase. This phase is responsible for the determination of the regenerator location. LERP has an added function/step that optimizes the regenerator placement.

[42] is another widely cited paper in the past three years of work done in this area of research, with 23 citations. Here the authors solve the offline IA-RWA for transparent networks. The authors present two types of algorithms –

1. An algorithm based on a global search which employs three different types of Binary Integer Linear Programming (BILP).
  - a) BILP-1: maximizes the number of lightpaths established, without considering any impairment constraints.
  - b) BILP-2: maximizes the Q-factor of a lightpaths without considering the non-linear impairments
  - c) BILP-3: minimizes the degradation in the Q-factor, due to the addition of new lightpaths in the network.
  
2. Two different types of heuristic solutions are proposed by the authors of [42]. After the pre-order phase, both algorithms run  $k$ -SP on each demand, to get  $k$  shortest paths between the source and the destination. A path that can be assigned a wavelength without affecting the quality of the other existing lightpaths is chosen. If no such path exists, then the demand is blocked. The two different approaches of ordering are given below:
  - a) Shortest Path First (SPF): Demand set is arranged in ascending order of the shortest path lengths between the source and the destination.
  - b) Longest Path First (LPF): Demand set is arranged in descending order of the shortest path lengths between the source and the destination.

## Chapter 3

# DESIGN AND IMPLEMENTATION OF THE TEST-BED

### 3.1 Overview

This chapter includes the description of the work done in the thesis. This chapter describes a test-bed that can execute any number of algorithms developed to solve the static IA-RWA problem for transparent optical network and compare their performances. The main motivation behind this work is the lack of a general test-bed, which has the capability of running any new algorithm for IA-RWA and then comprehensively compare the performance of the new algorithm to existing algorithms, based on some common criteria. This test-bed is a simulator, which in computer science essentially means a machine (or software) with the capability of creating an artificial environment under which software can be executed to get some diagnostic results that will allow the testers/researchers to analyze the software's performance in real-life.

The major building blocks of the simulator are shown in Figure 3.1. In the figure, the block called the *Simulator* is the most important part of the block diagram, as all other components interact with the simulator for their functioning. The simulator includes the following capabilities:

1. Generating a random network topology and an appropriate random demand set (data-set).

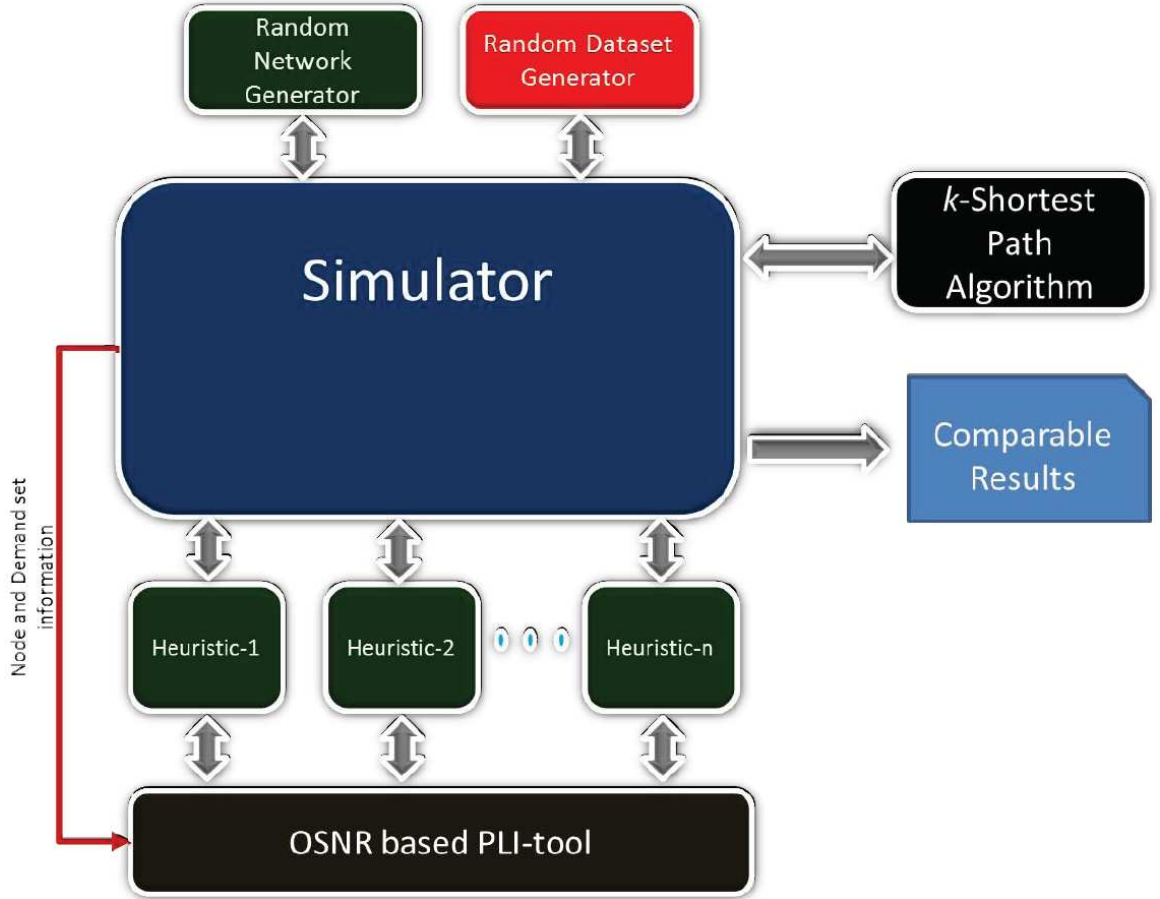


Figure 3.1: Block diagram for the “Simulator”

2. Calling the heuristic function(s) in a predetermined sequence to carry out IA-RWA on the data-set.
3. Printing the result.

All other components of the test-bed can interact with each other with the help of the simulator. The simulator can hence be termed as the *core* of the tool.

The Algorithm 3.1 gives an abstract working details of the test-bed. The input to the test-bed are data structures (virtual containers containing data). These data structures then are passed on to various components of the test-bed for initialization of the data. This data is finally used by the heuristics for their proper execution. As an output, the test-bed prints the result received from the execution of the heuristic.

A *network topology* can be programmatically represented by an adjacency matrix. The matrix in (3.1) shows the adjacency matrix for the graph shown in Figure 3.2. If a graph has  $N$  nodes, then an adjacency matrix will have  $N \times N$  elements. A cell in the matrix is represented by the notation  $[r, c]$ , where  $r$  ( $1 \leq r \leq N$ ) represents the row index and  $c$  ( $1 \leq c \leq N$ ) represents the column

---

**Algorithm 3.1** Algorithm followed by the Simulator
 

---

**Require:** Uninitialized Network information, Data-set information and  
Heuristic-List

**Ensure:** Result File with data comparing the performance of each heuristic

```

1: network_set_filenames ← generate_random_networks(number_of_NS)
2: for all (network_set ∈ network_set_filenames) do
3:   network_info ← initialize_network_information(network_set)
4:   dataset_filenames ← generate_random_dataset(network_set)
5:   for all (dataset ∈ dataset_filenames) do
6:     dataset_info ← initialize_dataset_information(dataset)
7:     heuristic_list ← initialize_heuristic_list()
8:     for all (heuristic ∈ heuristic_list) do
9:       result ← run_heuristic(heuristic, network_info, dataset_info)
10:      result_list ← include_in_list(result_list, result)
11:    end for
12:  end for
13:  for all (result ∈ result_list) do
14:    print_result(result)
15:  end for
16: end for

```

---

index. Hence, an edge between two nodes can be represented by the values given to a cell, e.g. for the cell  $[1, 4]$ , the value given is **95**, which essentially represents the edge from node  $E_1$  to  $E_4$  with a distance **95**. The algorithm shown in Algorithm 3.1 begins with the generation of a random network topology. The adjacency matrix can be stored in files. In Algorithm 3.1, `network_set_filenames` is the list of file names that carry the information. In Figure 3.1, the *random network generator* (RNG) block takes care of this responsibility. In the algorithm, the `network_info` data structure carries all the information about the network. For each `network_set`, a `network_info` is initialized with network topology information.

$$\begin{bmatrix} 0 & 0 & 0 & 95 & 35 & 0 \\ 0 & 0 & 0 & 50 & 0 & 95 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30 & 55 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 90 & 0 & 0 & 0 \end{bmatrix} \quad (3.1)$$

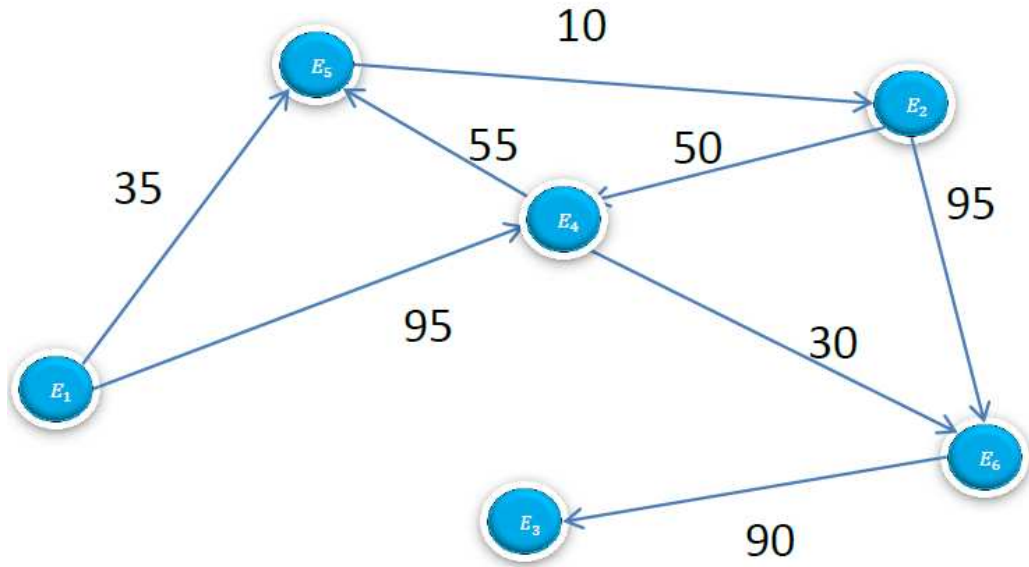


Figure 3.2: A simple network

This work uses the terms demand set and data-set interchangeably. Similar to the representation of the network topology, a demand set is represented using a matrix with dimension  $\text{NO\_OF\_SD} \times 2$ , where  $\text{NO\_OF\_SD}$  is the number of source and destination pairs to be generated. The matrix shown in (3.2) gives the matrix representation of a typical data-set with four demands. If a row index is represented by  $r$  then the cell  $[r, 1]$  represents the source node and the cell  $[r, 2]$ ,

represents the destination node of a demand. In the Algorithm 3.1 the next step is to populate the data-set files with the random demands. In Figure 3.1, the block called *random dataset generator* (RDG) assumes the above responsibility. The `data_set_info` data structure carries all the information about the demand set. For each demand set the `data_set_info` data structure is then initialized.

$$\begin{bmatrix} 5 & 6 \\ 1 & 6 \\ 4 & 3 \\ 6 & 1 \end{bmatrix} \quad (3.2)$$

In Algorithm 3.1, the `heuristic_list` is initialized with the locations of all the heuristic algorithms. In Figure 3.1, corresponding blocks to the `heuristic_list` data structure, are given the name “Heuristic 1”, “Heuristic 2” ... “Heuristic n”. The test-bed runs each `heuristic` available in the list, by passing the `network_info` and `data_set_info` as an input to the `run_heuristic` which executes each `heuristic` to return with a `result`, which is stored in the data structure called `result_list`. For each result in the `result_list`, the result is printed into a result file by the procedure `print_result`. The block called “Comparator”, represents the output for the test-bed. This block lists the number of successfully established demands for each heuristics and hence facilitates the user to draw out different conclusions based on the test-bed configurations. As the heuristics solve the issue of a static Impairment Aware RWA, they need a tool to calculate the extent of the impairments incurred into the network due to various routing and wavelength assignment decisions made during the execution of a heuristic. In Figure 3.1, this measure of the impairments is calculated using the block called *OSNR based PLI tool* (OPT).

The rest of the chapter is organized as follows. In Section 3.2, a detailed discussion on implementation of the test-bed and its components is given. Section 3.3 gives a detailed insight on three heuristics that were implemented in this thesis to test the working of the test-bed.

## 3.2 The Test-bed

This Section will discuss the detailed implementation, working and interaction of various components of the test-bed. This thesis will take the assistance of *Unified Modeling Language* (UML), to discuss how the various components of the test-bed work. UML is a universally accepted language for the software design blueprints [63]. The interaction and working of various components of software is better defined by a sequence diagram. A sequence diagram is part of a UML

interaction diagram, that shows the input and output events associated with the system under analysis [63]. Interaction diagram and sequence diagram will be used interchangeably.

An important point to be noted is that when the components are shown as “interacting” (or “communicating”) with each other, this is an abstract way of describing the communication between different services or functions in the actual software, which essentially occurs through global files and data structures (virtual containers containing data).

### 3.2.1 The Simulator

As mentioned earlier (Section 3.1), the simulator block forms the core component of the work done in the thesis. The other components interact with each other through the simulator block. The simulator executes an appropriate function within a component to get some result back. This result, if required, will be passed as an input to the other components’ functions. By controlling the sequence of execution, the simulator is able to make the decision of which component needs to execute next. The main goal of the simulator block is to execute all the heuristic algorithms and finally print out a result which reflects the performance of each heuristic in terms of their blocking ratio (blocking probability).

Some of the parameters that will be used in this Section are given below. These parameters can be changed by the users according to their preferences:

**NO\_OF\_NODES** The number of nodes in a graph.

**NO\_OF\_NS** The number of network sets. A network set is defined by various properties of a particular network. For example, number of edges, weights assigned to the edges and degree of each node (number of edges incident on each node).

**NO\_OF\_DS** The number of demand sets per network set.

**NO\_OF\_SD** The number of demands per demand set.



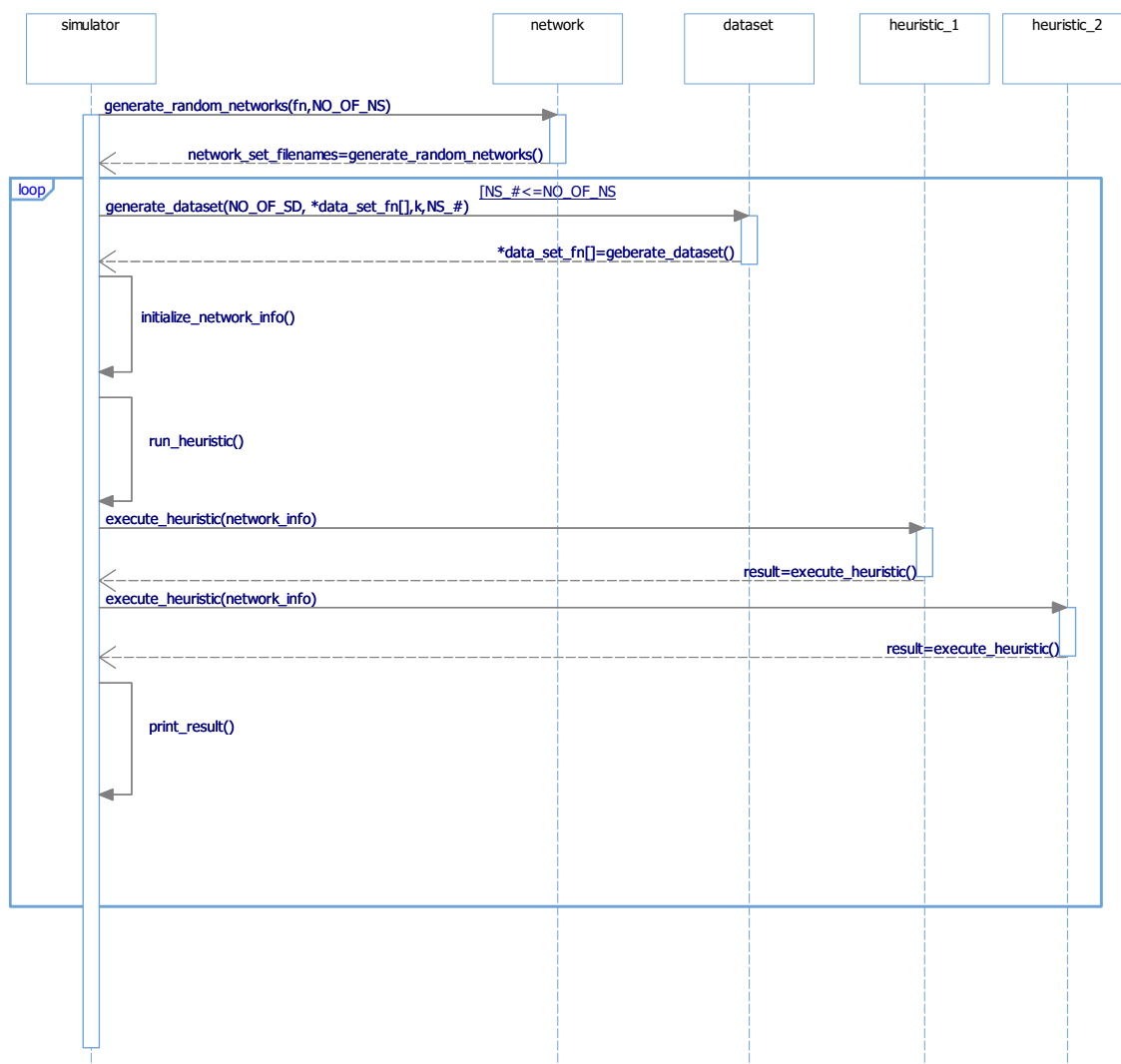


Figure 3.3: Simulator interaction diagram

Figure 3.3 shows the interaction diagram illustrating detailed working of the simulator block. In this diagram, only two heuristics for IA-RWA have to be executed. Assuming that the parameter `NO_OF_NS` is carrying the right information, the simulator is responsible for generating the random network (of size  $\text{NO\_OF\_NODES} \times \text{NO\_OF\_NODES}$ ) by using the service (`generate_random_networks`) provided by `network` object (the RNG). The RNG or the `network` object requires the simulator to provide it with an empty container (or data structure representing the filenames of the random networks) and the parameter `NO_OF_NS`. The parameter `NO_OF_NS` decides the number of networks the RNG has to generate. When the process is over, the simulator gets the location of the files containing the network information.

The next service to be accessed is the `generate_dataset` service provided by the `data-set` object or RDG. The service requires the simulator to provide the RDG with an empty data structure representing a collection of file-

names that will contain the list of demands. Along with data structure it needs some additional data like the network set number (**NS\_#**) and a parameter (**k**) representing the minimum number of shortest paths between the source and destination generated by RDG (refer to Section 3.2.3 for more details). The RDG randomly generates **NO\_OF\_DS** number of data-sets and each of these data-set contains **NO\_OF\_SD** randomly generated demands and returns with the list of file-names that contains the information.

A heuristic implementation has to satisfy two requirement defined by the **simulator** object, which allows the **heuristic** object to get the required parameters from the **simulator** for its execution and then returns a result, interpretable by the **simulator**. One of the global data structures that allow the communication between the heuristic and simulator is **network\_info**, which essentially is the information about the network that a heuristic will need for its execution. After the accesses to the services of RNG and RDG, this data structure is populated by an internal service (**initialize\_network\_info**) present in the **simulator**. The data structure contains the following information:

**topology** the network graph.

**demand-set** also known as data-set, which is a collection of traffic demands.

Another global data structure is **result**. Each heuristic is expected to return this data structure that essentially contains the name of the heuristic and the number of demands that have been successfully established by the heuristic. The receipt of the result from the first heuristic indicates the successful execution of the heuristic. Hence, the simulator then passes the same information (**network\_info**) to the next heuristic (**heuristic\_2**). The parameter **NO\_OF\_ALGORITHMS**, indicates the number of heuristic available for execution. The end of the last heuristic's execution acts as a trigger to access another internal service (**print\_result**) which prints the results received from the execution of different heuristics.

The stages after the random network generation, namely, the random generation of data set, the execution of the heuristic and finally the printing of the results from the execution of the heuristic algorithm, are repeated for each network set. The simulator therefore generates one result file for each network set.

### 3.2.2 The Dataset Generator

The Random Dataset Generator (RDG) plays an important role of generating the demand sets that are sent as an input to the heuristic algorithms. The number of datasets to be generated for every network topology is determined by the parameter **NO\_OF\_DS**. All the data sets and the network topology information for which

the demands are generated become part of the data structure `network_info`. This data structure is passed as an input to the heuristic algorithm. Many topologies do not allow the generation of a large number of shortest paths. To preclude such situations, only networks which have a sufficient number of shortest paths between any source and destination were considered. A parameter called `MIN_NO_OF_K` specifies the minimum number of shortest paths between any source and any destination. The dataset generation tool has two other configurable parameters which determine the distribution of source-destination pairs generated. The parameters `PERCENTAGE_UPPER_BOUND` and `PERCENTAGE_LOWER_BOUND` are responsible for setting the upper and the lower bounds of the randomly generated source-destination pair lengths. For example, if the lower limit is set to 20% of the optical reach and the upper limit is set to 80% of the optical reach then any source-destination pair whose shortest path length does not fall within the specified range will be discarded.

### 3.2.3 The Network Topology Generator

The Random Network Generator (RNG) is another important component of the test-bed that supplies the heuristics with the correct set of inputs. The RNG is responsible for generating a connected graph where the degree of each node lies between a specified lower and upper bound of nodal degree. The parameters that determine the structure of the network topology are the number of nodes, maximum and minimum nodal degree, the number of source and destination requested and finally the upper and lower bound for the edge length (randomly generated). The number of such topologies generated is determined by the parameter `NO_OF_NS`.

### 3.2.4 The $k$ -Shortest Path Algorithm

As indicated earlier (Section 2.4), the  $k$ -shortest path ( $k$ -SP) algorithm finds the first  $k$  shortest paths between any two vertices in a given graph. It was also indicated (Section 2.3.2) that a network can be conveniently represented by a graph. Hence, using the  $k$ -SP on a graph representing a network can result in first  $k$  shortest paths between any two nodes in the network. In this thesis the  $k$ -SP algorithm presented in [34] has been implemented. Since this algorithm plays a very important role in the implementation of the test-bed, this Section is dedicated to the discussion of the approach employed to implement the  $k$ -SP algorithm in [34]. One important parameter in deciding the extent of linear impairments in the optical networks is the optical reach. The implementation of the  $k$ -SP algorithm in this thesis hence considers the optical reach as a constraint

when generating the  $k$  shortest paths.

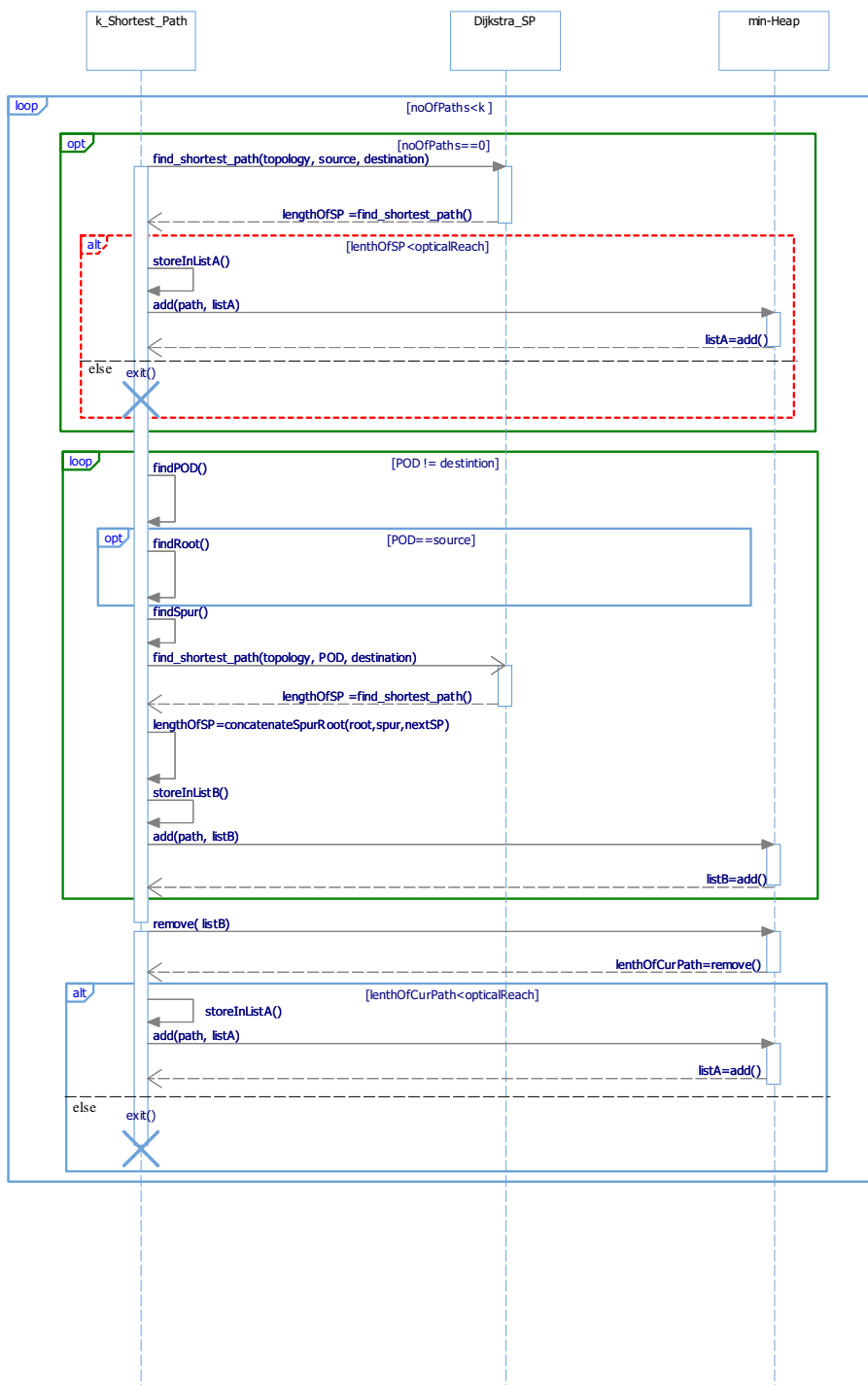


Figure 3.4: An implementation of  $k$ -SP

As shown in Figure 3.4, the implementation of  $k$ -SP was achieved by the collaboration of various components explained as follows:

1. **Dijkstra’s Shortest Path Algorithm** Given a graph with  $N$  nodes, Dijkstra’s shortest path algorithm [38] essentially finds the shortest path to

all the vertices from a desired vertex. In Figure 3.4, it is represented by `Dijkstra_SP` object. Dijkstra's algorithm is a labeling algorithm, since every node is labeled with the tuple  $(i, d)$  where  $1 \leq i \leq N$  represents the immediate predecessor of the node for the path to the node from the source node and  $d$  is the shortest distance to reach the node. Initially, distances to all the nodes are marked as infinity. Hence, the tuple for each node except the source node appears as  $(-, \infty)$ , meaning that the node is not been visited yet, and the shortest path from the source node to this node is also not determined. Dijkstra's algorithm is explained with the help of an example. The graph shown in Figure 3.5 is a bidirectional graph and the labels assigned to the edges of the graph represent the distance between the two nodes. A node is said to be "complete" when a shortest distance to the node from the source node is found.

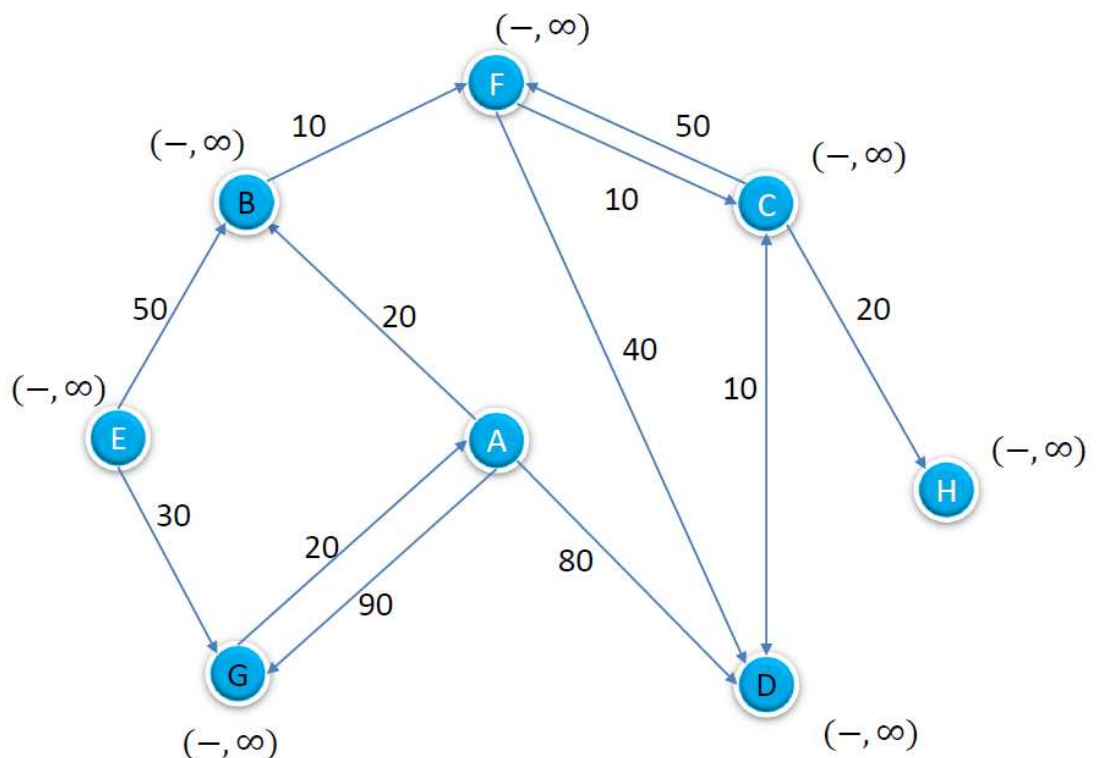


Figure 3.5: An example network

Let the source vertex (node) be  $A$ . As indicated earlier, initially, all nodes in the graph are labeled with tuple  $(-, \infty)$ . The algorithm first finds the distances to the nodes that are reachable in one hop. Figure 3.6 shows the

nodes shaded with a solid-diamond pattern, which have a direct edge from  $A$ . Since any distance value is an improvement over a value of infinity, the labels of nodes  $B$ ,  $D$  and  $G$  are now changed to  $(A, 20)$ ,  $(A, 80)$  and  $(A, 90)$  respectively. Figure 3.7 now shows the next step of the algorithm, in which it chooses the node with the shortest length and marks it as “complete”. In this case, the node  $B$  is marked “complete”. The next nodes to explore will be through the node which is “complete” and has the smallest value. The nodes marked complete are shaded with diagonal lines pattern.

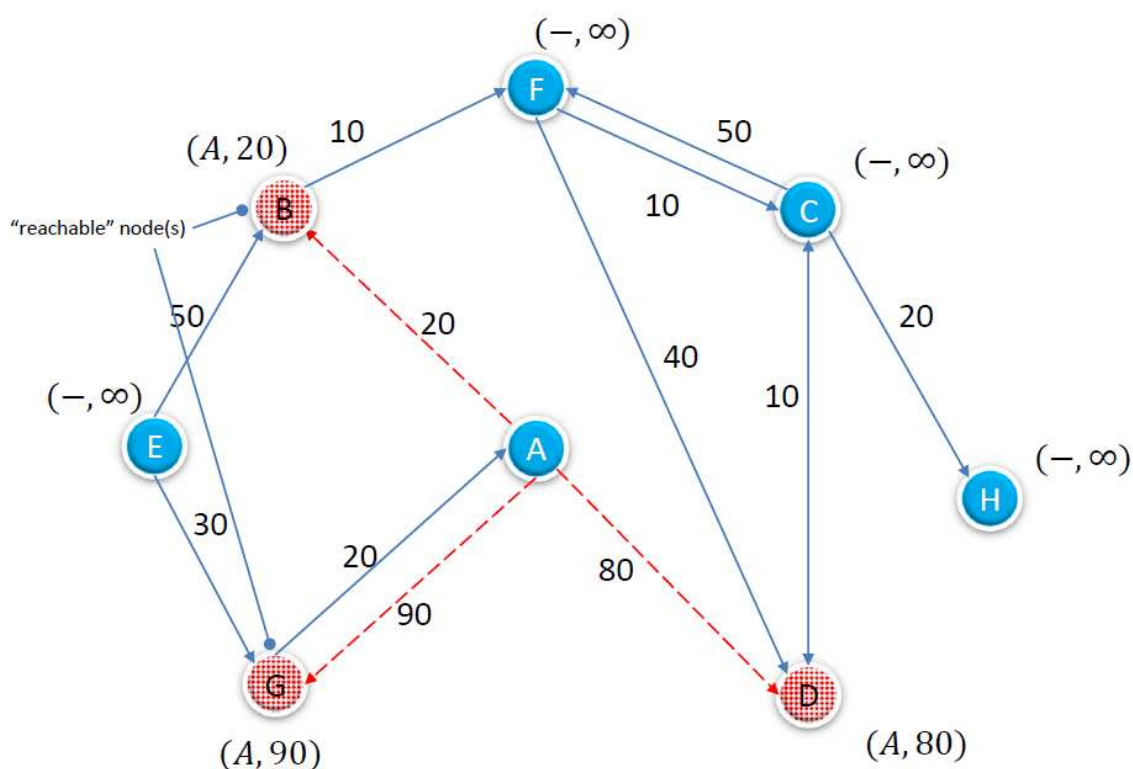


Figure 3.6:  $A$ 's direct neighbors are  $B$ ,  $G$  and  $D$ . Hence these are marked “reachable nodes”

As shown in Figure 3.8(A), the next node to explore is  $F$ , as  $B$  has just one outgoing node and accordingly, the label for  $F$  changes to  $(B, 30)$ , indicating that the node  $F$  can be reached through node  $B$ , with shortest distance (until now) **30**. Figures 3.8 show different self-explanatory transitions of the algorithm. The point to be noticed in Figure 3.8(B) is that the label for node  $D$  changes from  $(A, 80)$  to  $(F, 70)$  as the algorithm found a better route to reach the node  $D$ . Further in Figure 3.8(C), the node label changes to represent a better route to  $D$  (from  $A$ ).

Due to the repetition in the steps, Figure 3.9 directly shows the final result obtained. The first point to note from the final graph is that the node  $E$  is unreachable from node  $A$ .

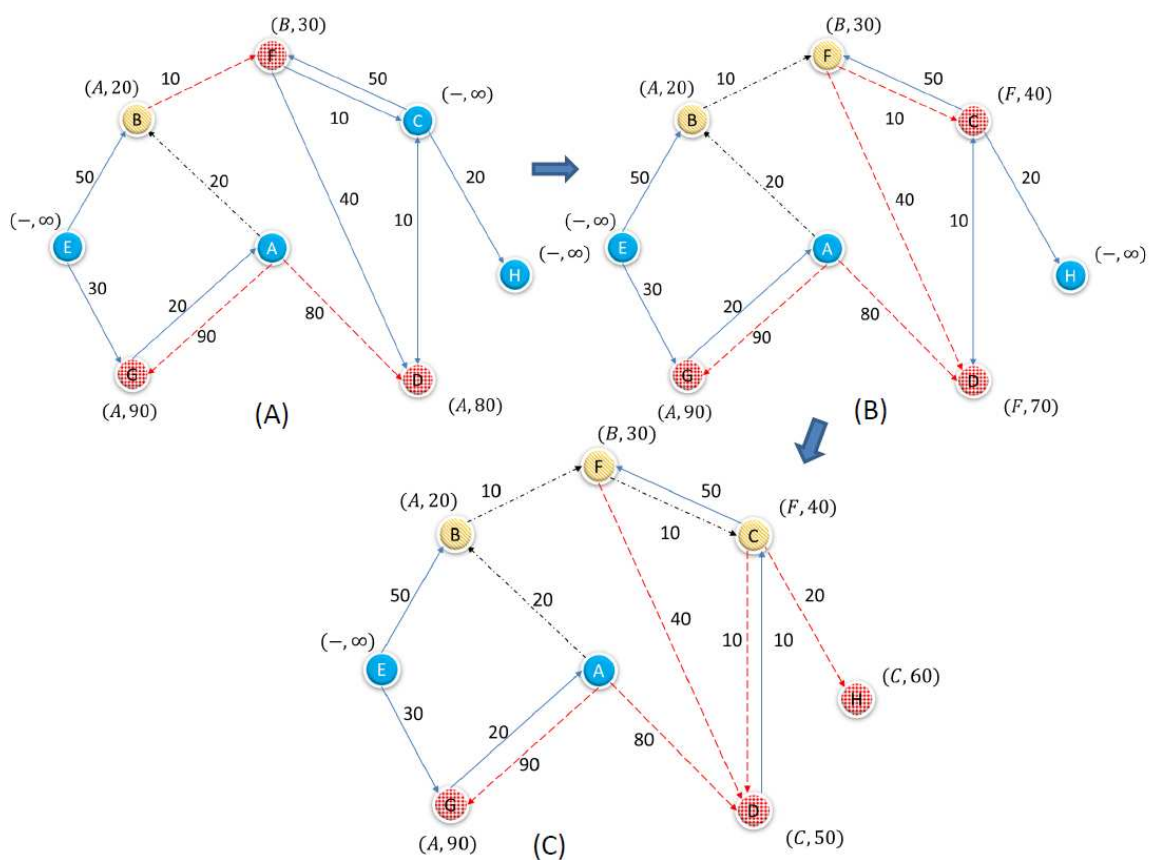


Figure 3.7: Different transitions a graph (or a network) goes through in Dijkstra's shortest path algorithm

Finally, the shortest path to any node can be found by following the label from the destination node to finally reach the source, e.g., to get to the node  $D$  through the shortest route, one has to reach to  $C$ . Similarly, the shortest route to  $C$  is through the route  $A \rightarrow B \rightarrow F$ , giving the resultant path as  $A \rightarrow B \rightarrow F \rightarrow C \rightarrow D$ , with length **50**.

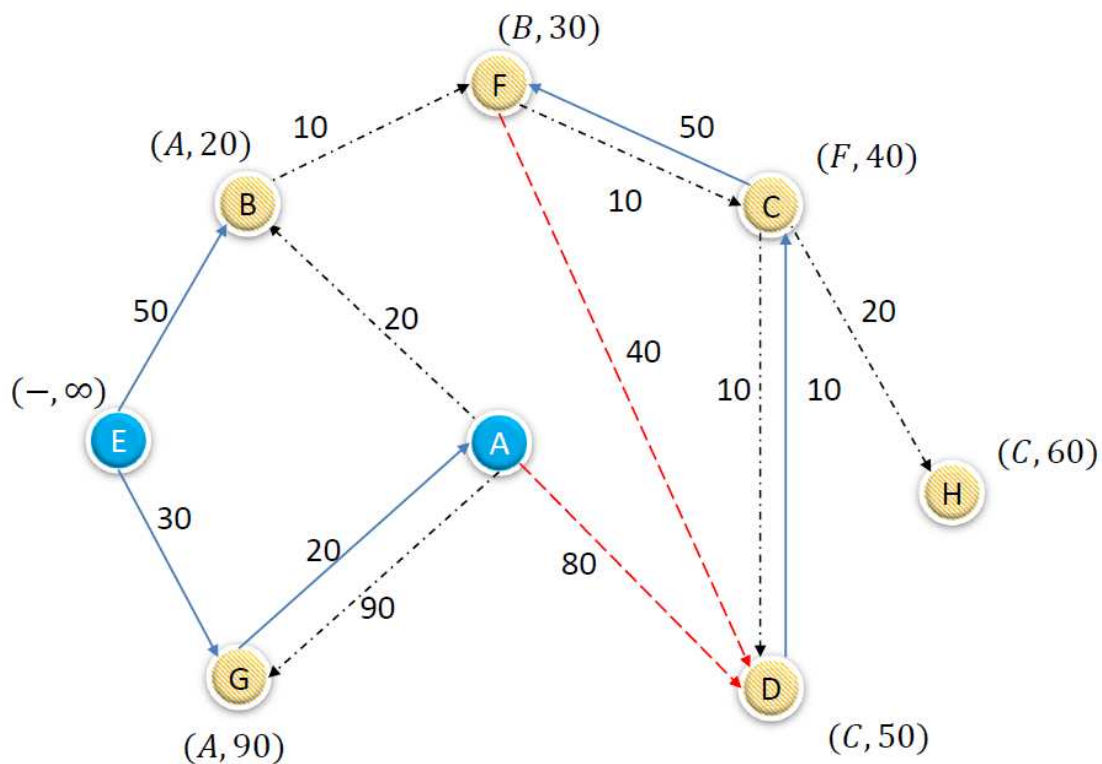


Figure 3.8: The final graph

2. **min-Heap** In graph theory a tree is an undirected graph where there exists a single simple path (a loopless path) between any two vertices. In computer science a tree can be used as a data structure to hold data and perform various operations on the data [64]. A *heap* is a type of tree-based data structure where a property known as heap property is satisfied [65]. The *heap property* states that if  $A$  is a parent node of some node  $B$ , then  $value(A) \geq value(B)$  if the heap is implemented as *max-Heap* otherwise the implementation is known as *min-Heap* where the condition to be followed is  $value(A) \leq value(B)$ . Any operation on the data structure has to maintain the heap property.



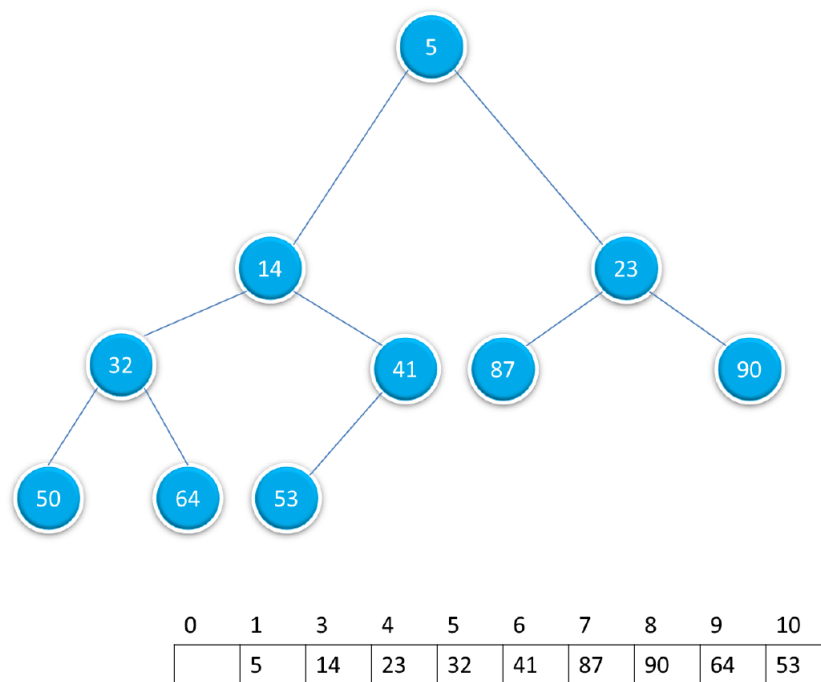


Figure 3.9: *min*-Heap example and its array implementation

The example shown in Figure 3.10 presents a *min*-Heap, along with its array implementation. Two important properties to highlight about the root of a *min*-Heap are that (i) it always has the minimum value of all the elements present in the heap, and (ii) it has an index of 1. For an element at index  $i$  in a *min*-Heap, the indices of its left child (if any), right child (if any) and parent (if any) are given by the formulae  $2i$ ,  $2i + 1$  and  $i/2$  respectively. In this thesis, the **min-Heap** is implemented as an auxiliary data structure, which will be used to store the candidate shortest paths.

- $k$ -Shortest Path** The  $k$ -SP implementation is presented in Figure 3.4. The figure shows how the `k_Shortest_Path` object collaborates with the `Dijkstra_SP` and the `min-Heap` object to generate  $k$  shortest paths. The two lists that are responsible to store the shortest paths and the “candidate” shortest paths are termed as `listA` and `listB` respectively. The two lists are implemented as *min*-Heap and thus any operation on these lists will result in the call of the services provided by the `min-Heap` object. This means that when adding (using the service `add`) any path to the lists, the paths have to follow the heap property (mentioned above) for a *min*-Heap. The `remove` operation removes the path pointed at by the index 1, hence returning the path with the minimum length.

As indicated before, the optical reach is considered to be an important parameter for the determination of the linear impairments in the IA-RWA

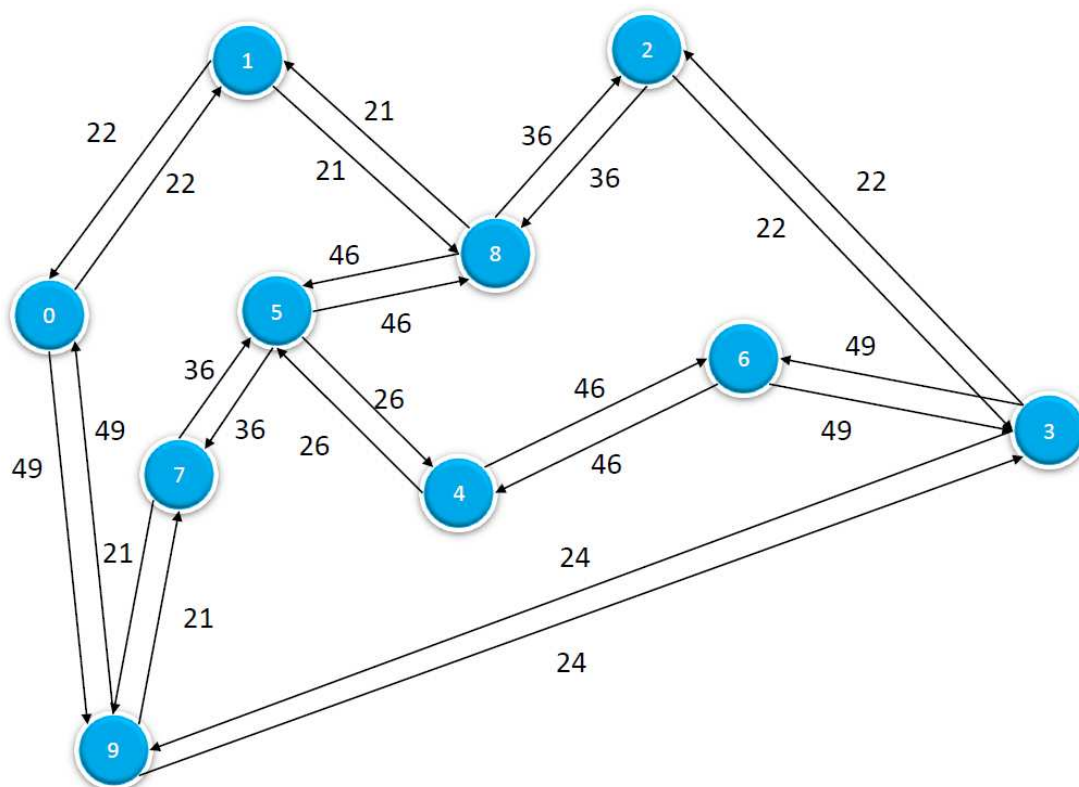


Figure 3.10: An 10-node network

algorithms. In the case of IA-RWA for transparent optical networks, paths that have lengths longer than the optical reach will be automatically deleted by the heuristic solutions. Since, there is no point in generating such paths. This additional constraint has to be introduced in the implementation of the  $k$ -SP algorithm. In Figure 3.4, the outer loop determines the number of iterations required to produce the desired result. The operation inside the loop will generate the next shortest path. The two parameters controlling the loop are `noOfPaths`, which is constantly updated to keep track of the number of paths generated and the `lengthOfCurPath`, which is constantly updated with the length of the last shortest path generated. The algorithm ends if either `noOfPaths`  $\geq k$  ( $k$  = number of shortest paths requested) or `lengthOfCurPath`  $\geq$  `optical_reach`.

In Section 2.4.2, it was indicated that the algorithm in [34] goes through two stages of iterations. Correspondingly, in Figure 3.4, there are two stages. One of the frames is an *opt* frame, the execution of which is controlled by a condition. In this case, if there are no paths generated the first frame is executed. The first shortest path is found by accessing the `find_shortest_path` service, provided by the `Dijkstra_SP` object. This service returns the length of the path that has just been found. Otherwise,

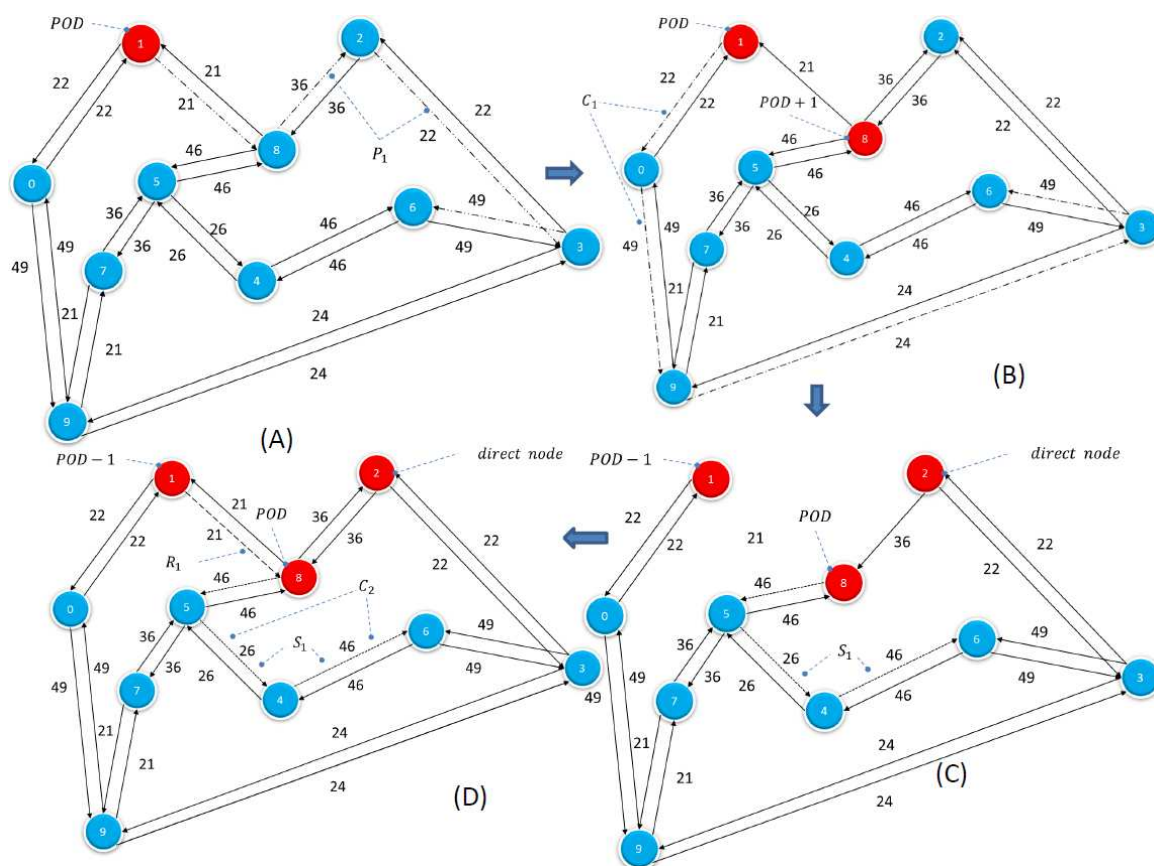


Figure 3.11: Different steps in the  $k$ -SP

if the path length is smaller than the optical reach then the algorithm simply exits. The last frame makes sure that if the length of the next generated shortest path is greater than or equal to the optical reach, the algorithm simply stops. The next frame is a loop, which runs until the “Point of Deviation” (refer to Section 2.4.2 for a precise definition) or  $POD=destination$ . A network of  $N = 10$  nodes, shown in Figure 3.11 is used as an example to explain the next few steps involved in this iteration. As an example, let source node ( $s$ ) = 1 and the destination node ( $d$ ) = 6, the number of shortest paths to be found ( $k$ ) = 2 and the optical reach be 1000 units. The first shortest path ( $P_1$ ) is found using the Dijkstra’s Shortest Path Algorithm is  $1 \rightarrow 8 \rightarrow 2 \rightarrow 3 \rightarrow 6$  with length **128**. With the assistance of the aforementioned example, the steps involved in the iteration are explained as follows:

**findPOD()**– this service finds the next Point of Deviation (POD). The first POD is the first node of the last shortest path found ( $P_1$ ). At the end of the iteration, the POD is updated to point to the next-SP node in the path. In Figure 3.12 (A) the path  $P_1$  and the POD is clearly indicated.

**findRoot()** – this service finds the “Root” sub-path in the last found short-

est path. This sub-path is part of the shortest path from the source to POD. As indicated in the sequence diagram, this part of the step is not required when the `POD=source`, as the candidate shortest path can be found without taking this step into account (see `findSpur()`).

`findSpur()` – this service is responsible for finding the “Spur” (refer to Section 2.4.2 for precise definition) from the POD, which is a key to find the candidate shortest path. Let an edge from node  $i$  to node  $j$  (such that  $0 \leq i, j \leq N$ ) in the topology be represented by  $(i, j)$ . This step involves more sub-steps as mentioned below:

- a) If the `POD=source`, then edge  $(\text{POD}, \text{POD}+1)$  is deleted from the topology. A shortest path from the node `POD` to the node `destination` is found by accessing the `find_shortest_path` service in the `Dijkstra_SP` object. The next step after this is `concatenateSpurRoot()`. Figure 3.12 (B) depicts this step, where the edge  $(1, 8)$  is removed to get the path  $1 \rightarrow 0 \rightarrow 9 \rightarrow 3 \rightarrow 6$  with length **144**. As this path is the whole path from the source to the destination, it will be stored in the `listB` as a “candidate” shortest path ( $C_1$ ) by calling an internal service `storeInlistB`, which in turn calls the service `add` in `min-Heap`.
- b) If the `POD!=source` then the algorithm removes the direct edges from the `POD` to the nodes that have been visited already in the paths stored in `listA` and the node `POD-1` (this part ensures generation of loopless path) from the topology. The “Spur” is found by searching for the shortest path from the `POD` to the `destination` on the modified topology. In Figure 3.12 (C) the edges  $(1, 8)$ ,  $(8, 1)$  and  $(8, 2)$  are deleted to get the spur ( $S_1$ ) -  $8 \rightarrow 5 \rightarrow 4 \rightarrow 6$ .

`concatenateSpurRoot()` – this step of the iteration concatenates the “Root” and the “Spur” sub-path, found in the previous steps. Figure 3.12(D) illustrates this step. The  $R_1$  and  $S_1$  are the “Root” and “Spur” for the `POD = 8`. The concatenation of these two sub-paths generates the next “candidate” shortest path  $C_2$ , which is stored in `listB` by following the step mentioned previously (see `findSpur()` step (a)).

The candidate paths found in the second stage of the  $k$ -SP are summarized in the Table 3.1 In this table the shortest among the “candidate” paths is the one with length 139. Hence, the path is removed from the `listB` to insert into the `listA` signifying the promotion of the “candidate” shortest path to “the next” shortest path.

Table 3.1: The “Candidate” shortest paths. *NF* =Not Found

POD	POD - 1	Root	Spur	Candidate Path	Length
1	N/A	N/A	1→ 0 →9→ 3→ 6	1→ 0→ 9→ 3→ 6	144
8	1	1	8 →5 →4→ 6	1→ 8 →5 →4→ 6	139
2	8	1→ 8	<i>NF</i>	N/A	N/A
3	2	1 →8 →2	3→ 9→ 7→ 5→ 4→ 6	1→ 8 →2→ 3→ 9 →7→ 5→ 4→ 6	232

In the above example it should be noted that the  $k$ -SP algorithm would not have considered “the next” shortest path and would have exited, if the length had been greater than or equal to the optical reach.

### 3.3 Implementation of the Heuristic

This section presents three heuristics that have been implemented to test the validity of the test-bed. The executions of all the three heuristics are controlled by the test-bed. The test-bed is responsible for providing the heuristic with valid inputs. The test-bed assumes that each heuristic follows the specified protocols for the test-bed. Once the heuristic successfully executes, it returns with a data structure which essentially represents the number of demands that have been accepted by the heuristic. Section 3.3.1 describes the implementation of the classical RWA with  $k$  shortest paths, which does not consider any PLI. Sections 3.3.2 and 3.3.3 present two different heuristic approaches that considered PLIs that are implemented in this thesis. All of the above listed Sections use the topology in Figure 3.12 to illustrate the working of the corresponding heuristic.

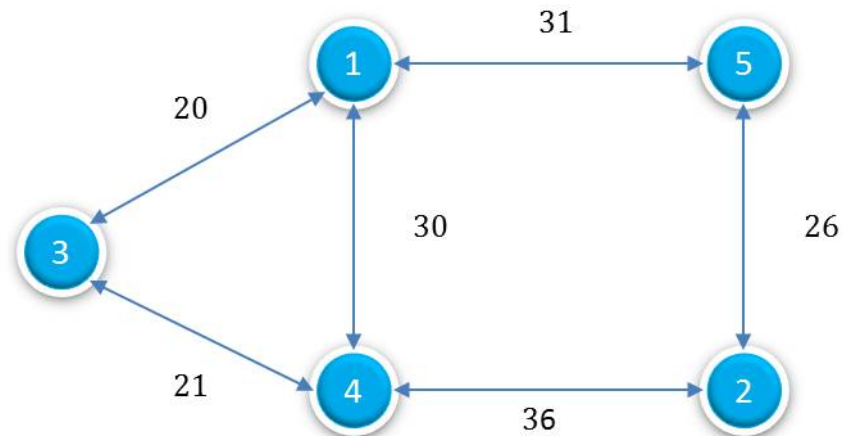


Figure 3.12: Example topology used for illustration of the heuristics

### 3.3.1 Classical RWA (cRWA) with $k$ -Shortest Paths

This thesis implements the classical routing and wavelength algorithm while considering  $k$  shortest paths in the routing part of the algorithm. This type of RWA is termed *seq*-RWA by the authors of [31].

#### Objective

The objective of this algorithm is to find  $k$  shortest paths for each demand and sequentially assign an available channel to one of the  $k$ -SP, for each demand. During the wavelength assignment phase, the wavelength continuity constraint is taken into account.

#### Approach

To implement the algorithm presented in [31], the *seq*-RWA algorithm can be divided into two parts. The first part finds the  $k$  routes using the  $k$ -SP implementation presented in Section 3.2.4. The second part is responsible for sequentially assigning a channel to one of the feasible  $k$  shortest paths for each demand. While assigning the channels, the wavelength continuity constraint needs to be taken care of. For simplicity, if any of the source-destination is repeated, it is treated exactly like any other demand.

#### The Algorithm

The implementation of the algorithm is described with the help of a small example. Following are the parameters considered for the example:

$$N \text{ (Number of Nodes)} = 5$$

$$C \text{ (Number of Channels per Fiber)} = 3$$

$$D \text{ (Number of Demands)} = 4$$

$$k = 4$$

$NF$       The next shortest path between the source and the destination is Not Found. This happens due to two reasons: (i) the length of the next shortest path has exceeded the optical reach, or (ii) there is no shortest path available

The demands are shown in Table 3.2.:

Table 3.2: Demands with corresponding  $k$ -SPs

Source	Destination	$k$ Shortest Paths
4	1	$4 \rightarrow 1 \rightarrow 3$ $4 \rightarrow 0 \rightarrow 3$ $4 \rightarrow 0 \rightarrow 2 \rightarrow 3$ $NF$
0	4	$0 \rightarrow 4$ $0 \rightarrow 3 \rightarrow 1 \rightarrow 4$ $0 \rightarrow 2 \rightarrow 3 \rightarrow$ $1 \rightarrow 4$ $NF$
1	2	$1 \rightarrow 3 \rightarrow 2$ $1 \rightarrow 4 \rightarrow 0 \rightarrow 2$ $1 \rightarrow 3 \rightarrow 0 \rightarrow 2$ $1 \rightarrow 4 \rightarrow 0 \rightarrow$ $3 \rightarrow 2$
2	0	$2 \rightarrow 0$ $2 \rightarrow 3 \rightarrow 0$ $2 \rightarrow 3 \rightarrow 1 \rightarrow$ $4 \rightarrow 0$ $NF$

The algorithm goes as follows –

**Routing Phase** For each couple of nodes in the demand set, this phase finds  $k$  shortest paths. The algorithm presented in [34] is implemented to find the  $k$ -SPs (Section 3.2.4).

### Channel Assignment Phase

1. For each demand in the demand set, select the path  $P_i$  (for all  $i, 0 \leq i \leq k$ ).
2. For each edge in the path assign a path-free channel [31]. A path-free channel is a channel which has not been assigned to any other lightpath using this edge. Each edge in the path should have the same channel assigned. This ensures that the wavelength continuity constraint is enforced.
3. If there does not exist any such wavelength for the path, then select the next path  $P_{i+1}$ , if available. If no other path exists, mark the demand as blocked and go to step 1.

4. Repeat steps 1 to 3 until all the demands are considered.

Using the example stated above, the working of the algorithm is described below

–

1. Table 3.2 shows the source and the destination for each demand. It also gives the first phase of the algorithm by stating the  $k = 4$  shortest paths between each source and destination pairs. Due to the small size of the network, for most source-destination pairs, the  $k$ -SP algorithm is not able to find all 4 shortest paths.
2. Table 3.3 shows the summary of the channel assignment phase.

Table 3.3: Summary of channel assignment phase.

Demand #	Channel Selected	Path # selected	The path
1	0	0	4 → 1 → 3
2	0	0	0 → 4
3	1	0	1 → 3 → 2
4	0	2	2 → 3 → 1 → 4 → 0

The first two demands are trivially assigned the channel 0, as there are no edges that appear in both the paths. For the third demand, the edge from the node 1 to 3 has already been assigned the channel 0, hence the next “path-free” channel (i.e. channel 1) is selected for the path. In the case of the fourth demand, only the last path can be assigned with the “path-free” channel 0.

### 3.3.2 sLERP

sLERP, or simple Lightpath Establishment with Regenerator Placement [31], is a heuristic algorithm that solves the static IA-RWA problem in translucent optical network. Since this thesis only considers the transparent optical network, the implementation of this algorithm will not include the regenerator placement problem stated by the authors of [31]. This implementation uses the OPT (OSNR based PLI Tool) (Section 3.1) for testing the quality of a candidate lightpath.

#### Objective

The objective of sLERP is to get the list of candidate lightpaths by using the *Random Search based* (RS-based) algorithm and then test the quality of each of



these candidate lightpaths using the OPT. The candidate lightpaths that fail the OPT test are eventually rejected.

### Approach

To implement the algorithm presented in [31], the whole algorithm is divided into two phases. The first phase, called the *RWA phase*, uses the RS-based algorithm to find the candidate lightpaths. The next phase, termed as *Quality of Transmission Test phase*, takes the list of candidate lightpaths generated in the first phase as an input. Each lightpath in the list is sequentially tested for PLIs using the OPT. The lightpaths that do not maintain an acceptable quality all through its path are rejected.

### The Algorithm

The algorithm is explained with the assistance of the following example:

$$N \text{ (Number of Nodes)} = 5$$

$$C \text{ (Number of Channels per Fiber)} = 1$$

$$D \text{ (Number of Demands)} = 8$$

$$k = 4$$

$$MAX\_PERM \text{ (Number of permutations)} = 5$$

Table 3.4: Demands with corresponding  $k$ -SPs

Source	Destination	$k$ Shortest Paths
0	2	$4 \rightarrow 1 \rightarrow 3$ $4 \rightarrow 0 \rightarrow 3$ $4 \rightarrow 0 \rightarrow 2 \rightarrow 3$ <i>NF</i>
0	1	$0 \rightarrow 4 \rightarrow 1$ $0 \rightarrow 3 \rightarrow 1$ $0 \rightarrow 2 \rightarrow 3 \rightarrow 1$ <i>NF</i>
2	0	$2 \rightarrow 0$ $2 \rightarrow 3 \rightarrow 0$ $2 \rightarrow 3 \rightarrow 1 \rightarrow$ $4 \rightarrow 0$ <i>NF</i>

4	0	4→0 4→1→3→0 4→1→3→2→0 <i>NF</i>
3	2	3→2 3→0→2 3→1→4→0→2 <i>NF</i>
4	1	4→1 4→0→3→1 4→0→2→3→1 <i>NF</i>
3	4	3→1→4 3→0→4 3→2→0→4 <i>NF</i>
1	4	1→4 1→3→0→4 1→3→2→0→4 <i>NF</i>

**RWA Phase** The RWA phase (or RS-based RWA) of the heuristic extends the *seq*-RWA to assign routes and wavelengths to each permutation of the demands in a permutation set. The RS-based RWA can be divided into two sub-phases discussed below:

1. ***Routing Phase***

This follows the same step as the routing phase of *seq*-RWA.

2. ***Permutation and Wavelength Assignment Phase***

This phase involves the following steps:

- a) The order of the demands in the demand set is rearranged in a random order and is stored in a vector  $\rho_i$  for all  $0 \leq i \leq MAX\_PERM$ .
- b) For each vector  $\rho_i$ , wavelengths are assigned to each demand by following the steps described in the wavelength assignment phase of *seq*-RWA.
- c) The vector with the least number of blocked demands is chosen from the permutation set. If there are more than one such vector, then the

tie is broken by choosing the vector which used the least number of channels.

**Quality of Transmission Test Phase** The feasibility of each demand is estimated using the OPT. If a demand is marked not feasible by the OPT, then the demand is blocked.

Using the example stated above, the algorithm is described below –

1. Table 3.4 shows the routing phase of the algorithm. Due to the small size of the network, for most of the source-destination pairs, the  $k$ -SP algorithm was not able to find all 4 shortest paths.
2. Table 3.5 shows the summary of the permutations and the wavelength assignment.

Table 3.5: Summary of the permutation and wavelength assignment

$i$ where $0 \leq i \leq MAX\_PERM$	$\rho_i$	Number of Calls Blocked	Number of Channels Used
0	7 4 2 6 3 1 0 5	0	1
1	1 6 0 3 5 2 7 4	2	1
2	3 5 1 7 2 0 4 6	0	1
3	7 1 5 0 3 6 2 4	2	1
4	6 0 2 5 7 3 4 1	1	1

It is quite apparent from the table that the zeroth and the second permutation vectors are the ones with no blocked demands. According to [31], the vector with the least number of channels used will act as the tie breaker in such cases. But since there is only one channel per fiber available, any one of the vector can be arbitrarily selected. In this case the zeroth permutation vector is chosen

3. The next step is to feed the chosen list of candidate lightpaths into the OPT for QoT Test phase. It was found in this case that all the lightpath pass the test. Hence, these lightpaths can be now established in the network.

### 3.3.3 Shortest (Longest) Path First

The authors of [42] presented two types of pre-ordered heuristics to solve the static IA-RWA problem in transparent optical network. In these types of heuristics

the demands in a demand set are ordered according to some criterion. In [42], the authors order the demands in the demand set according to the length of the shortest path between the source and the destination in the demand. The implementation of the two types of pre-ordered heuristics is explained in the following paragraphs.

### Objective

The objective of the two heuristics presented in [42] is to prioritize the demands in ascending (or descending) order of the length of the shortest path of each demand and then perform a sequential routing and wavelength assignment simultaneously with quality analysis for each demand.

### Approach

The Shortest (Longest) Path First heuristic is divided into two phases. The first phase is responsible to come up with an ordered list of demands. A lightpath is considered *Q-feasible* if the establishment of such lightpath does not affect the quality of previously established lightpaths. The next phase sequentially solves the static IA-RWA problem by (i) generating  $k$  shortest paths for a demand, (ii) finding a wavelength for one of the  $k$  shortest paths and (iii) checking the *Q-feasibility* of the lightpath.

### The Algorithm

**Demand Preprocessing Ordering Module** Two types of ordering are considered:

1. In the case of the Shortest Path First (SPF), the demands in the demand set with lowest value of the length of the shortest path is selected first.
2. In the case of the Longest Path First (LPF), the demands in the demand set with lowest value of the length of the longest path is selected first.

### RWA and Q-Feasibility Test Phase

1. For each demand  $d_i$  (for all  $i, 0 \leq i \leq D$ , where  $D$  is the number of demands) in a demand set, search  $k$  shortest paths, using the algorithm implemented in Section 2.4.2.
2. For demand  $d_i$  choose a path  $P_j$  (for all  $j, 0 \leq j \leq k$ ) and find a wavelength  $c$  (for all  $c, 0 \leq c \leq C$ , where  $C$  is the number of channels per fiber) that is not used in any of the edges of  $P_j$  by previously established lightpaths.

This step is analogous to Step 2 of the wavelength assignment phase of the *seq*-RWA heuristic.

3. If such a wavelength is available then go to Step 3, otherwise choose the next path (if available)  $P_{j+1}$  for  $d_i$ .
4. Let step 2 produce a lightpath  $l_i$ . Check the feasibility of  $l_i$  using OPT. If the lightpath is marked infeasible by the OPT, then perform Steps 2 and 3 for  $P_{j+1}$  (if available) for  $d_i$ . If no other paths are available for demand  $d_i$ , mark  $d_i$  as blocked.
5. Repeat steps 1 to 4 until all the demands are considered.

# Chapter 4

## EXPERIMENTAL ANALYSIS

### 4.1 Overview

In this chapter, different experimental results are discussed. This chapter presents the approach adopted to study the effects of varying parameters relevant to RWA. The test-bed was used to execute all the heuristics. The test-bed allows users to study the performances of different heuristics for RWA under different scenarios. The experiments have been conducted basically on two types of networks – synthetic network topologies and wide-area network topologies which are currently in use. To generate the synthetic network topologies, RNG (described in Section 3.2) was used with different values of `NO_OF_NODES`. The values of `NO_OF_NODES`, in the experiments conducted ranged from 10 to 30. USANET [66], AT (American Topology) [67] and ARPANET [66] are the three different real-life network topologies used for the experiments. Different properties of the real network topologies considered are listed in Table 4.1

Table 4.1: Features of real networks

Network	Number of Nodes	Nodal Degree Range	Edge Length Range (km.)
AT	14	2-3	21-100
USANET	24	2-5	51-196
ARPANET	21	1-2	41-108

As mentioned before there are numerous parameters available in the test-bed that can be changed to evaluate the behavior of the heuristics that are being compared using the test-bed. Most of them will be discussed in detail in the the following Sections. The values assigned to some of the parameters are summarized in Table 4.2

Table 4.2: Some well known parameters

Parameter Name	Symbol	Value Range
Number of Channels	$\lambda$	(8,16,32)
Number of Nodes (for Synthetic Networks)	$N$	10-40
Number of Datasets	DS	20 (Fixed)
Number of Source Destination pair	$D$	50-300
Number of Heuristics	NO_OF_ALGORITHMS	1-4

As mentioned in Chapter 2, the PLI evaluating model, OPT (OSNR based PLI Tool), was incorporated in this thesis. Some of the OSNR constants considered in the model are summarized in Table 4.3. Technical details on the values of OSNR constants can be found in [67].

Table 4.3: OSNR constants

Constant Name	Value
Amplifier 1 Noise Factor	5 dB
Amplifier 2 Noise Factor	5 dB
Fiber Loss Co-efficient	0.2 dB/km
Optical Switched Loss	3 dB
Multiplexer Loss	3 dB
De-multiplexer Loss	3 dB
Switch Isolation Factor	-40 dB
OSNR at the starting node	30 dB
OSNR Threshold	23 dB
Input Optical Signal Power	-3 dBm
XPM loss in dB	-28 dB

One important point to note is that, in classical RWA with  $k$  shortest

paths (cRWA) technique, it is assumed that there exists an ideal physical layer, with no impairments. So, the results obtained through this heuristic represent the best-case scenario, and provide upper bounds on the number of successful connections for each scenario. In the experiments, heuristics cRWA, sLERP and SPF (LPF), described in Sections 3.3.1, 3.3.2 and 3.3.3, are compared against each other.

## 4.2 Synthetic Networks

In this chapter, the behaviors of different heuristics are studied under different synthetic network topologies. The test-bed provides numerous parameters that can be modified to observe the changes in the behaviors of the heuristics. In the following Sections the performances of the selected heuristics for the RWA were studied for various values of the following parameters:

1. Channels per Fiber ( $\lambda$ ).
2. Number of Nodes ( $N$ ).
3. Demands ( $D$ ).
4. Edge Length
5. Minimum Number ( $K$ ) of Shortest Paths between a Source-Destination Pair.
6.  $k$ -Shortest Paths (K\_PATHS).

The implementation of the heuristics considered in this Section is explained in the Section 3.3. Some of the values that are kept unchanged throughout the experiment (unless otherwise specified) are listed in Table 4.4.

Table 4.4: Constant parameters

Parameters	Values
Edge Length Lower Limit	20
Edge Length Upper Limit	40
Min Degree of a Node	2
Max Degree of a Node	3
DS	20
NO_OF_ALGORITHMS	4
Number of Network Sets (NS)	3



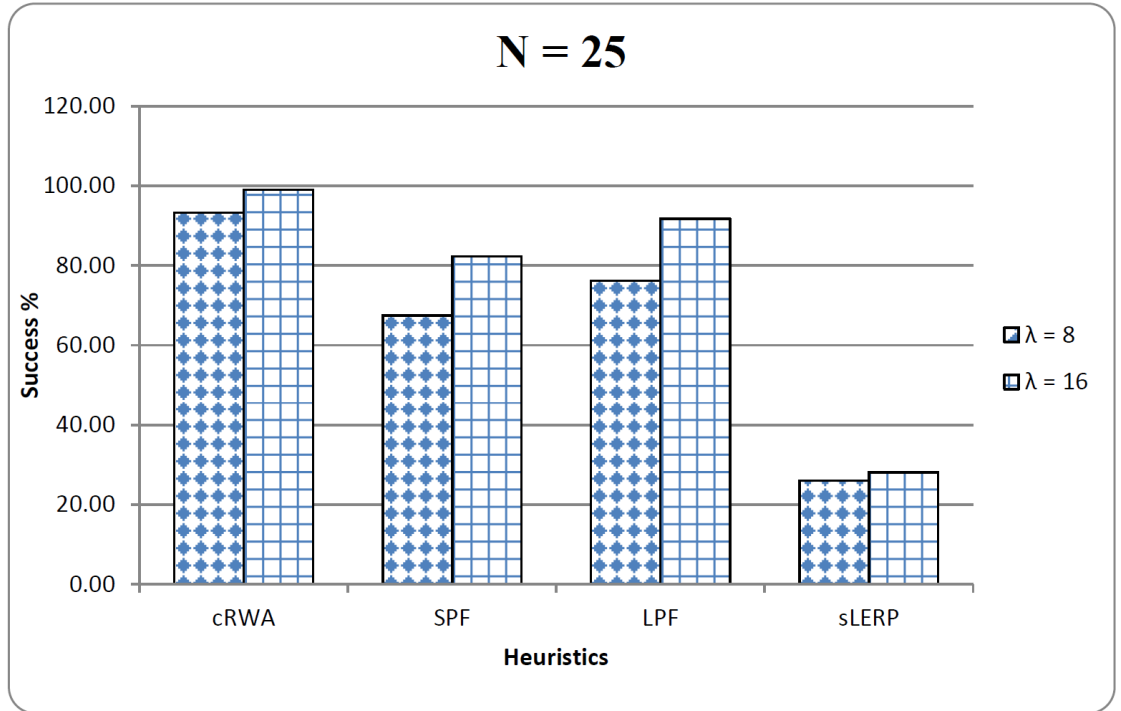


Figure 4.1: Synthetic network behavior with varying  $\lambda$

#### 4.2.1 Channels per Fiber ( $\lambda$ )

To study the behavior of the heuristic with varying  $\lambda$  values, the experiments were conducted with number of nodes  $N = 25$  and number of demands  $D$  assigned values ranging from 50 to 125. The overall success rate for a heuristic was obtained by determining the average percentage of number of successfully established demands for all network sets. The values for the number of network sets (NS) and number of demand sets (DS) can be found in Table 4.4.

The results were obtained for two values of  $\lambda$  (8 and 16). Figure 4.1 represents the average success rate obtained for all the demands in the network. The figure clearly shows the dominance of  $\lambda = 16$  over  $\lambda = 8$  for every heuristic. The number of channels per fiber, determines the availability of “path free” (Sections 3.3.1) channels for a route. The larger the number of channels per fiber higher will be the availability of such channels. As the number of such “path free” channels increases, possibility of a route been assigned a channel increases. In Figure 4.1, it can also be observed that the heuristic sLERP has the worst performance. The reason for this kind of behavior is discussed in Section 4.2.2.

### 4.2.2 Number of Nodes ( $N$ )

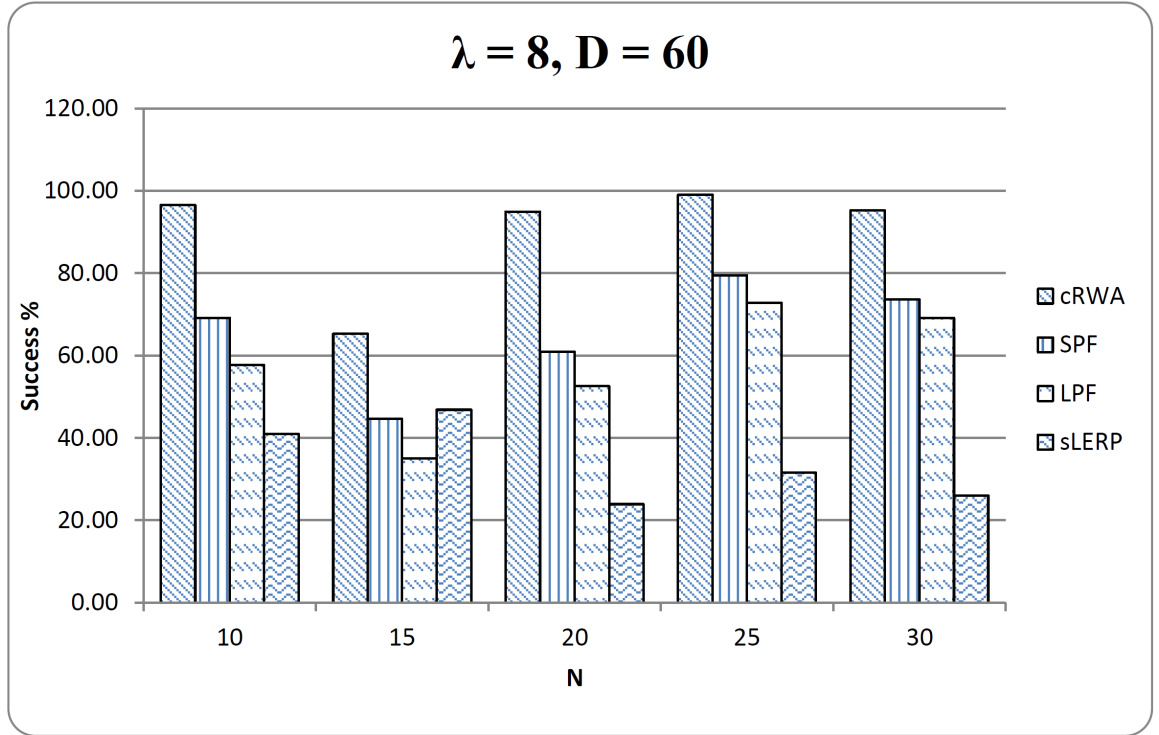


Figure 4.2: Varying  $N$  with  $\lambda = 8$

The study of heuristics also includes the effects of varying the value of the parameter  $N$ . The average success rate was determined for each type of network, for the number of channels per fiber,  $\lambda = 8$  and then  $\lambda = 16$ . Both Figure 4.2 and Figure 4.3 show that the behavior of the heuristics are similar, irrespective of the number of nodes in the network. All the heuristics are able to establish more lightpaths when  $\lambda = 16$  because of the reason stated in Section 4.2.1. As stated above, cRWA gives the best performance among all the heuristics. The performance of LPF (Longest Path First) is lower than the SPF (Shortest Path First), which conforms to the observation made in [42]. sLERP has the worst performance. Heuristic sLERP was originally designed to solve static IA-RWA in translucent networks, which essentially means that the implementation of sLERP in the thesis has ignored the regenerator placement problem and simply marks a demand “blocked” if a “path free” wavelength is not found. The removal of the regenerator placement problem from the heuristic hence increases the blocking probability of sLERP significantly.

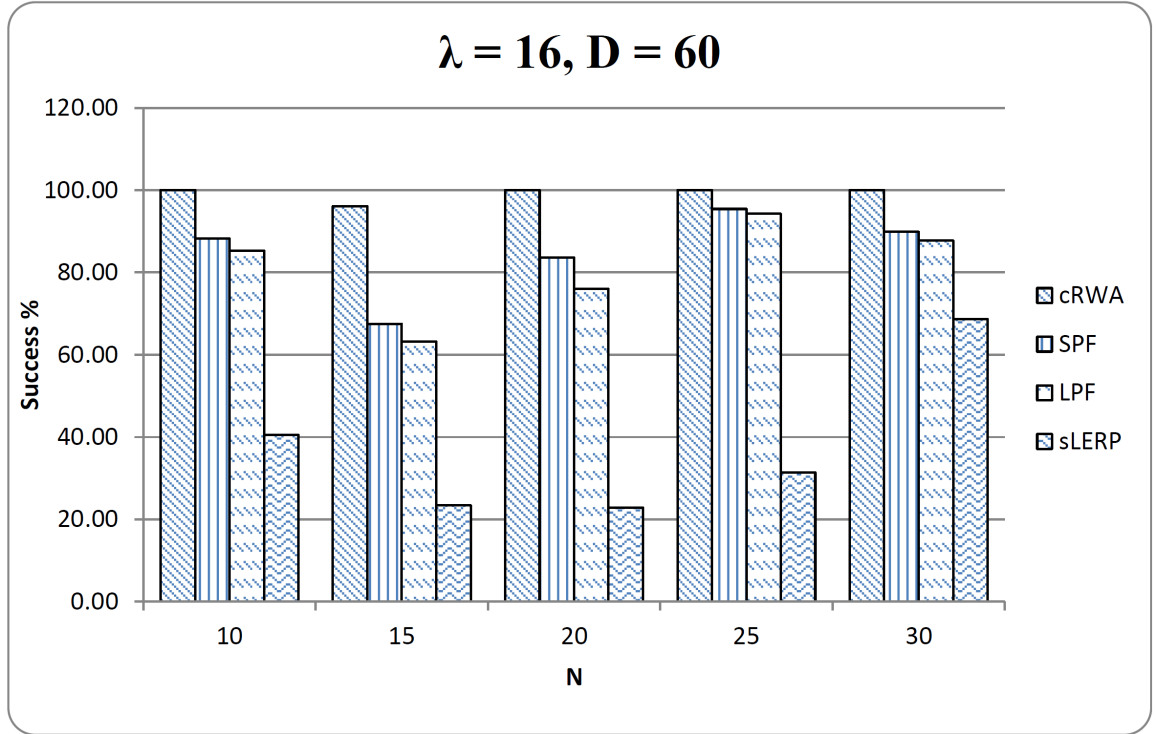
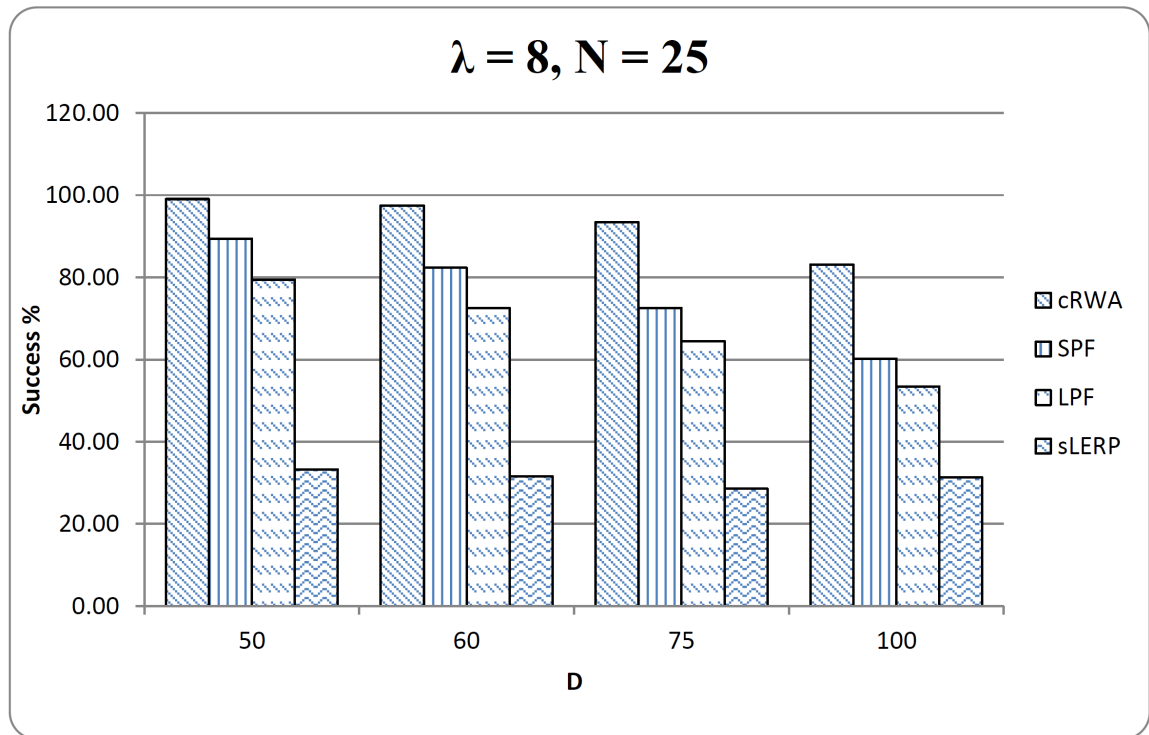
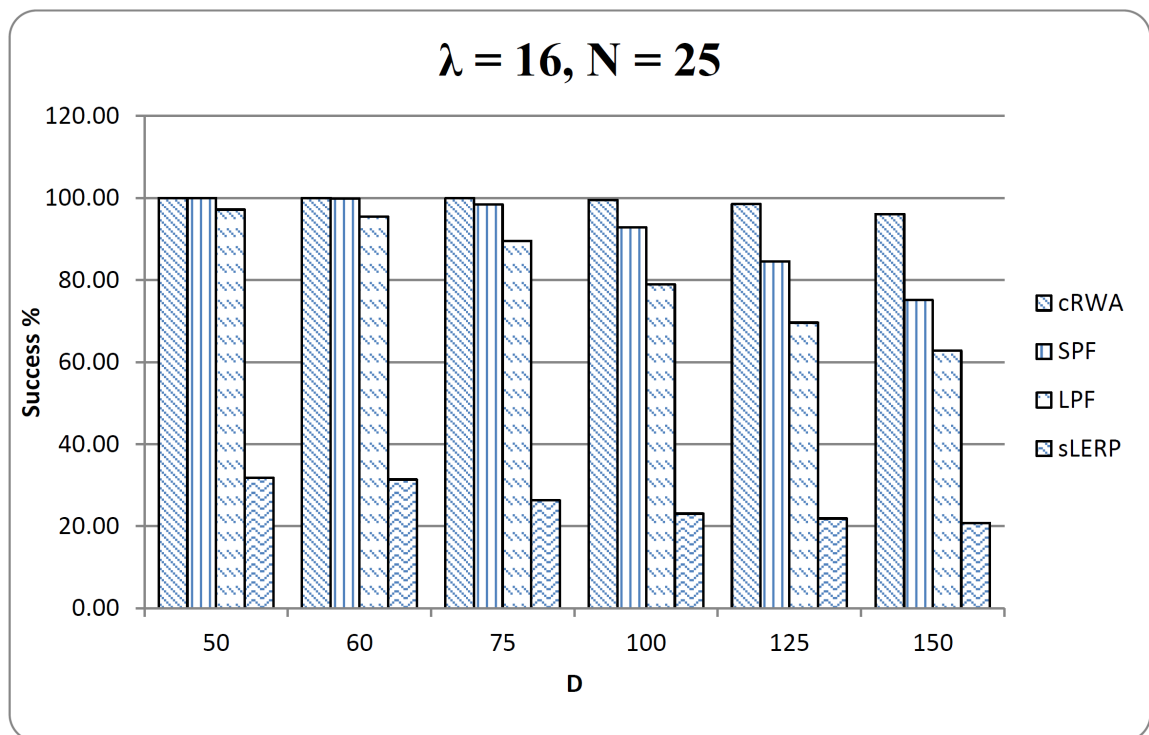


Figure 4.3: Varying  $N$  with  $\lambda = 16$

### 4.2.3 Demands ( $D$ )

In this Section the effect of varying the number of demands  $D$  for a particular type of network is studied. Here, the case where number of nodes  $N = 25$ , and the number of channels per fiber,  $\lambda = 8$  (and 16) is considered. From the two sets of results (Figure 4.4 and Figure 4.5) it may be observed that the network with  $\lambda = 16$  is able to support 9.59 % more demands than  $\lambda = 8$ . This is due to the reasons stated in Section 4.2.1. Further, for the value  $\lambda = 8$ , the network was not able to support a significant number of successful demands when  $D = 125$  and 150. The common trend that can be observed from both figures is that, as the number of demands on the network increases, the success rate decreases. This behavior is attributed from the fact that the increase in the number of demands increases the congestion on individual edges, and hence the success rate decreases.

Figure 4.4: Varying  $D$  with  $\lambda = 8$ Figure 4.5: Varying  $D$  with  $\lambda = 16$

#### 4.2.4 Edge Length

One of the important parameters that affects the linear impairment in the optical network is the length of the path a lightpath is traversing. If the edge lengths are too large then the routing part of the IA-RWA will result in longer routes (sometimes longer than the optical reach) and thus increase the number of blocked demands. This trend can be observed in Figures 4.6 and 4.7. For the experiments, the case where  $N = 10$ ,  $D = 60$  and  $\lambda = 8$  & 16 is considered. There are two parameters that control the length of an edge in a synthetic network, the values of `EDGE_LENGTH_LOWER_LIMIT` and `EDGE_LENGTH_UPPER_LIMIT`, stipulating the range for the length of an edge when randomly picking up a value for the edge length.

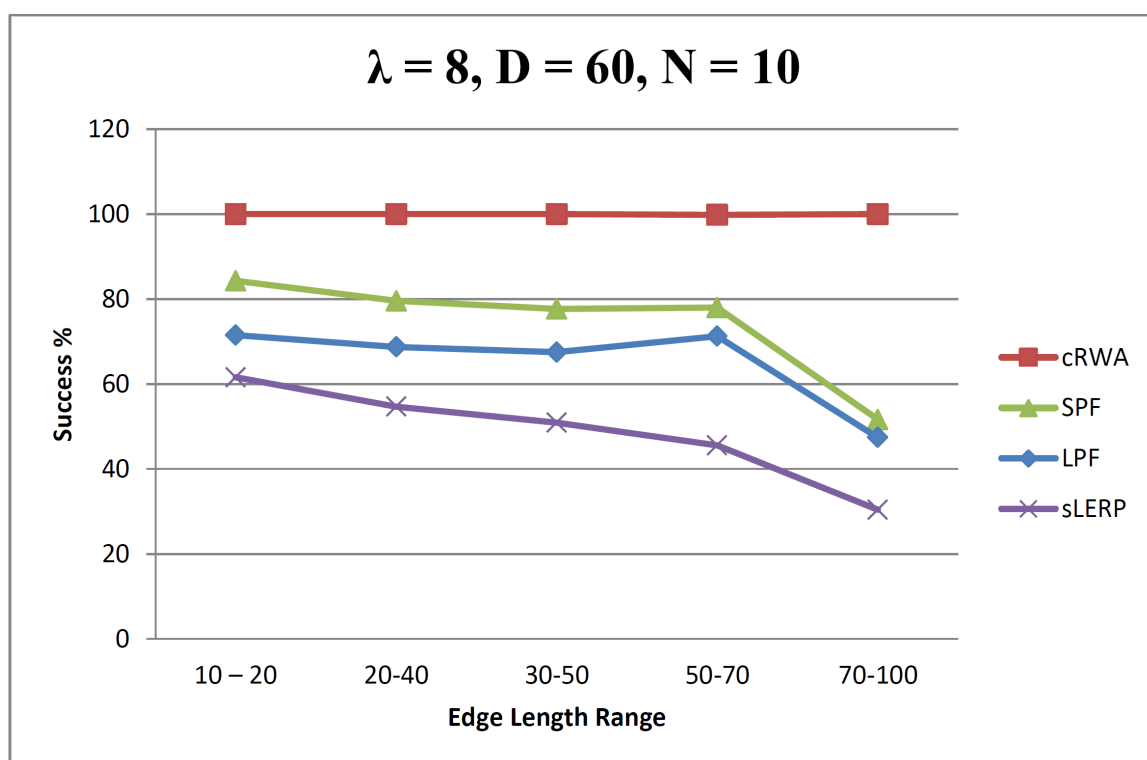


Figure 4.6: Varying Edge Length Lower Limit-Upper Limit with  $\lambda = 8$

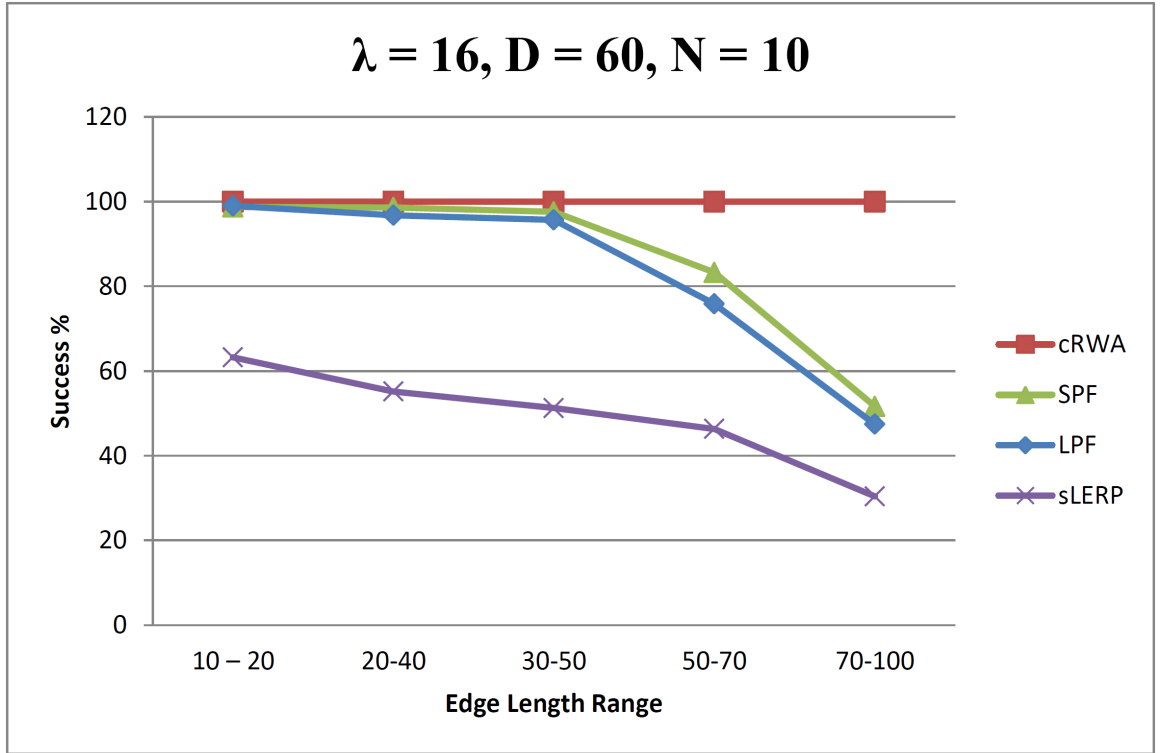


Figure 4.7: Varying Edge Length Lower Limit-Upper Limit with  $\lambda = 16$

#### 4.2.5 Minimum Number ( $K$ ) of Shortest Paths between a Source-Destination Pair

During the generation of random source and destination pairs, the parameter  $K$  determining the selection of a pair is the minimum number of shortest paths lying between the source and the destination node in the network. This parameter is also used by the Random Network Generator (RNG) package, while generating synthetic network topologies. Between each source and destination pair for a demand in the demand set, the RNG ensures that there are at least  $K$  shortest paths. In this section, the effect of varying the value of  $K$  is studied. For the experiment,  $D = 60$  and  $N = 30$  are maintained. The values for  $K$  are varied from 5 to 30 in increments of 5.

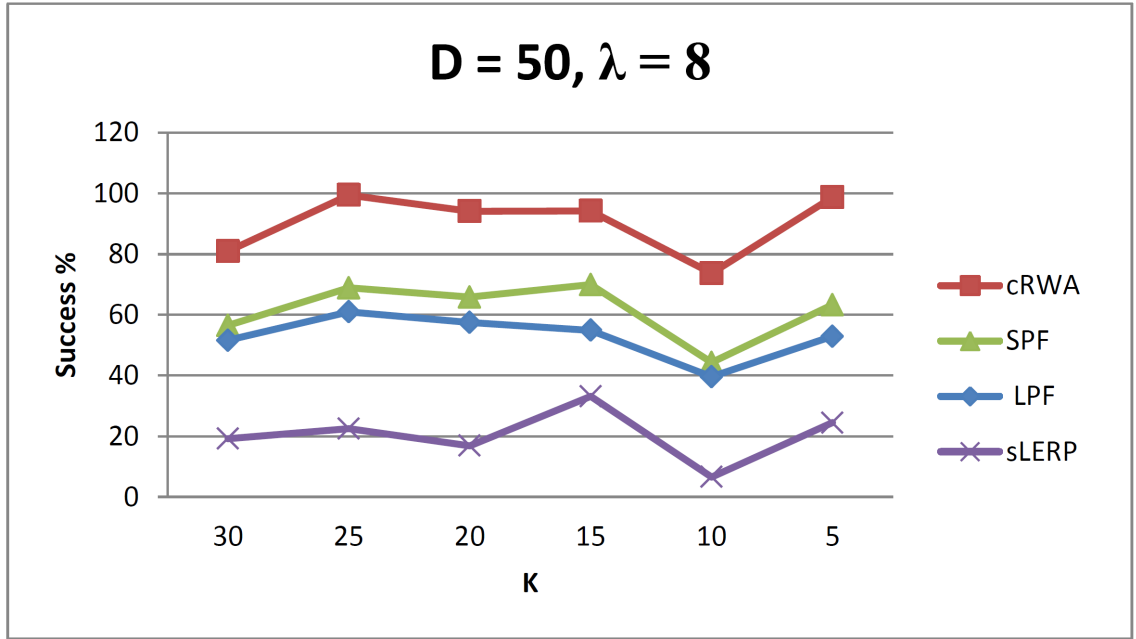


Figure 4.8: Success % for varying  $K$  values

From the above discussions, it is now clear how parameter  $K$  affects the generation of the synthetic network topology. The network topology generated when  $K = 30$  will be quite different from the network topology generated when  $K = 20$ . This is attributed to the fact that the topology that supports  $K = 20$  paths may not support  $K = 30$  paths. Hence, in Figure 4.8, no consistent behavior is observed.

#### 4.2.6 $k$ Shortest Paths ( $K\_PATHS$ )

In this experiment the behaviors of the heuristics are studied when the values of the parameter  $K\_PATHS$  is varied. An important point to note is that the parameters  $K$  and  $K\_PATHS$  are completely different parameters and are independent of each other. The parameter  $K$  decides the type of source and destination pairs selected by the test-bed. The value of  $K$  determines whether a network topology will be accepted. On the other hand the parameter  $K\_PATHS$  has no role to play in the demand or network topology generation. This parameter is used by the heuristics to decide the number of alternate shortest paths between a source and a destination pair that will be used, and if a path between the pair of nodes is rejected due to a network layer or physical layer constraint.

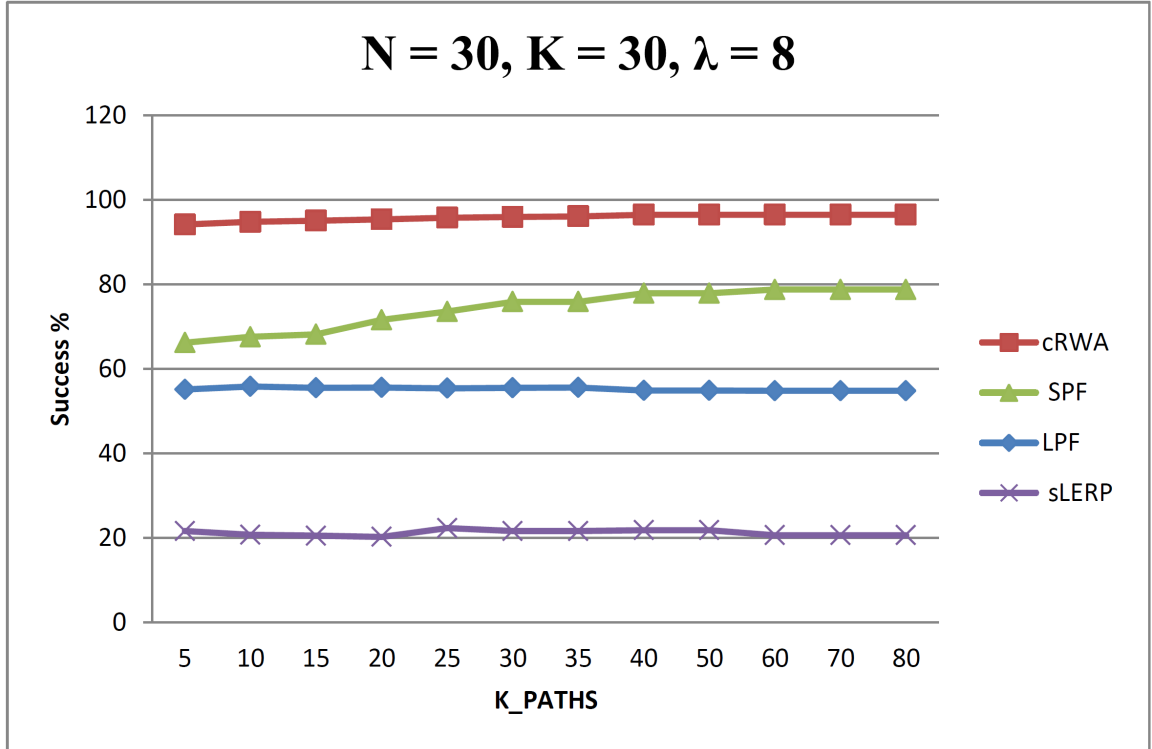


Figure 4.9: Varying K\_PATHS

In the experiment the parameters  $N$ ,  $K$  and  $\lambda$  assumed the values 30, 30 and 8 respectively. The K\_PATHS was varied systematically within the range 5 to 80. Figure 4.9 shows the different behaviors of the heuristics. The success rate for cRWA, sLERP and LPF, remain constant all along the varying values of K\_PATHS.

### 4.3 Real-life Networks

The synthetic networks allow a higher level of flexibility, since it is possible to vary a number of parameters such as the edge length and the number of nodes. In some cases it was also observed that, due to the random nature of some parameters, it was not feasible to control the behaviors of the heuristics (Sections 4.2.5). In this Section, a study of the heuristics is done by varying the values of some of the available parameters.

#### 4.3.1 American Topology

Figure 4.10 shows the American Topology (AT) with the distances assigned to each edge. The topology closely corresponds to NSFNET with reduced edge length and optimized for the OPT [67].



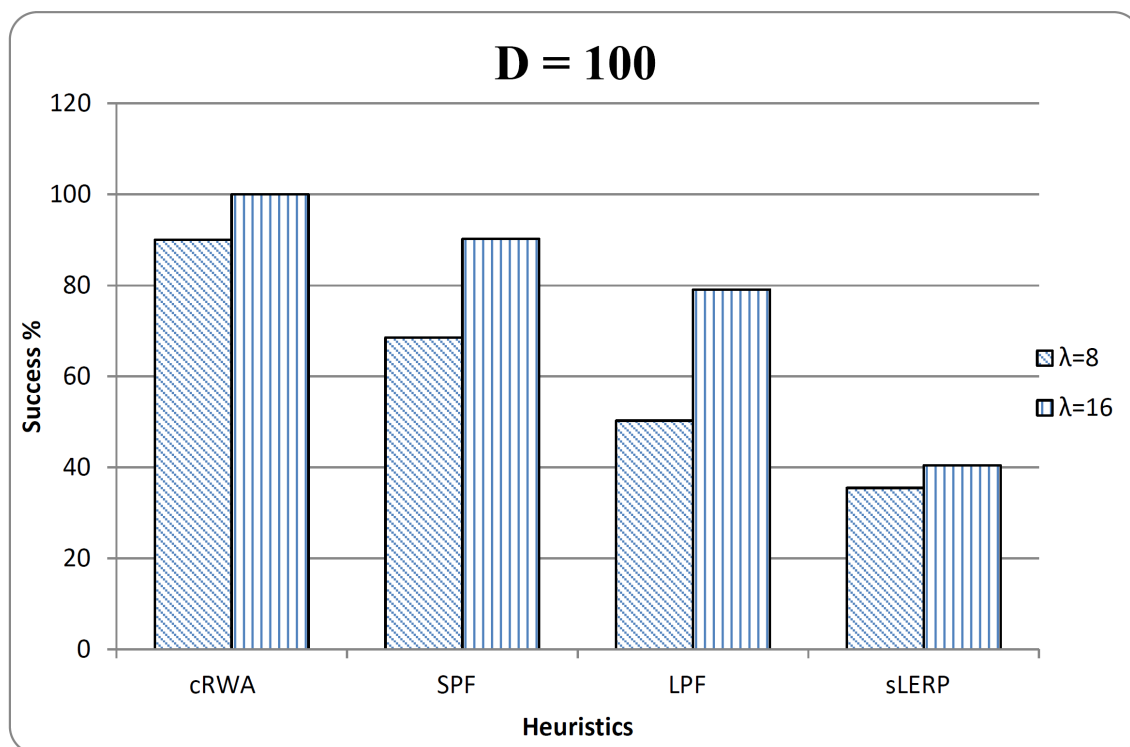


Figure 4.10: American Topology

#### 4.3.1.1 Channels per Fiber ( $\lambda$ )

The behaviors exhibited by the heuristics with respect to real-life networks are consistent with the behaviors observed in the case of synthetic networks when comparing their performance for the different values of  $\lambda$ . Figure 4.11 shows the summary of the results obtained with the number of demands  $D=100$ .

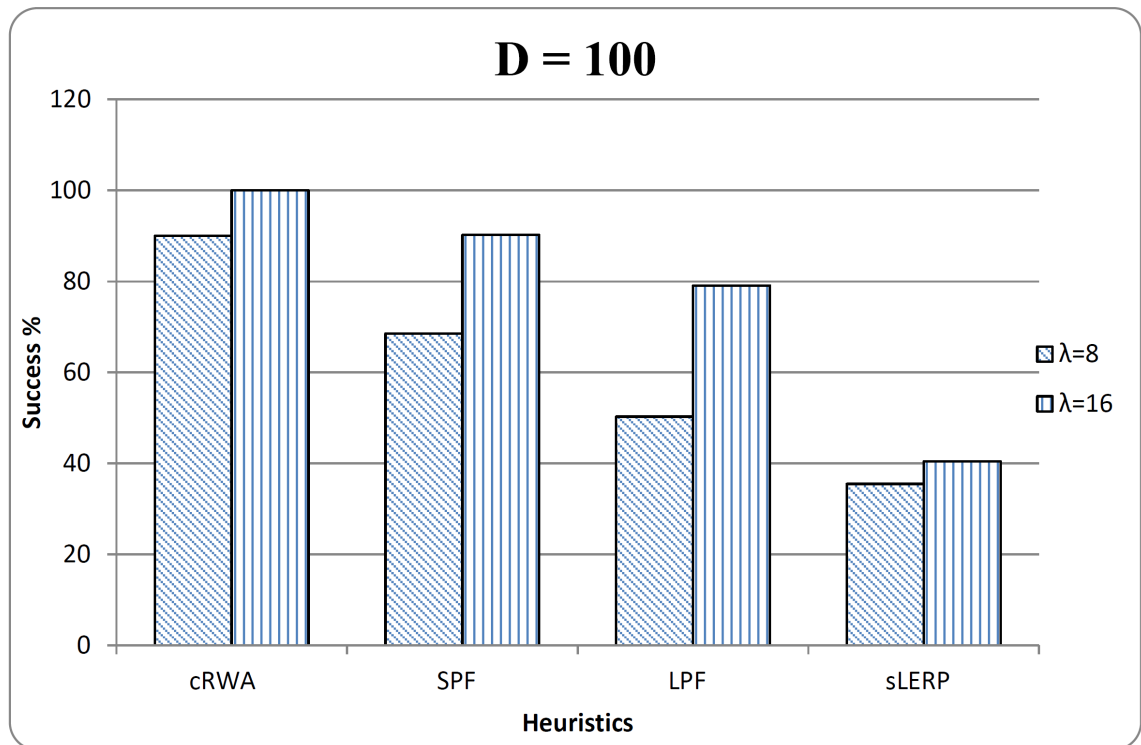


Figure 4.11: Effect of varying  $\lambda$  in AT

#### 4.3.1.2 Range for the Shortest Path

There are two parameters that determine the selection of a node pair as a member of a demand set, namely  $K$  and the PERCENTAGE\_UPPER\_BOUND ( $pub$ ) and PERCENTAGE\_LOWER\_BOUND ( $plb$ ) (Sections 3.2.2).  $plb$  and  $pub$ , together represents a range which control the selection of only those source and destination nodes which have a shortest distance lying between the values of  $plb$  and  $pub$ . As the AT was optimized specifically for the OPT, only this experiment is performed for this topology.

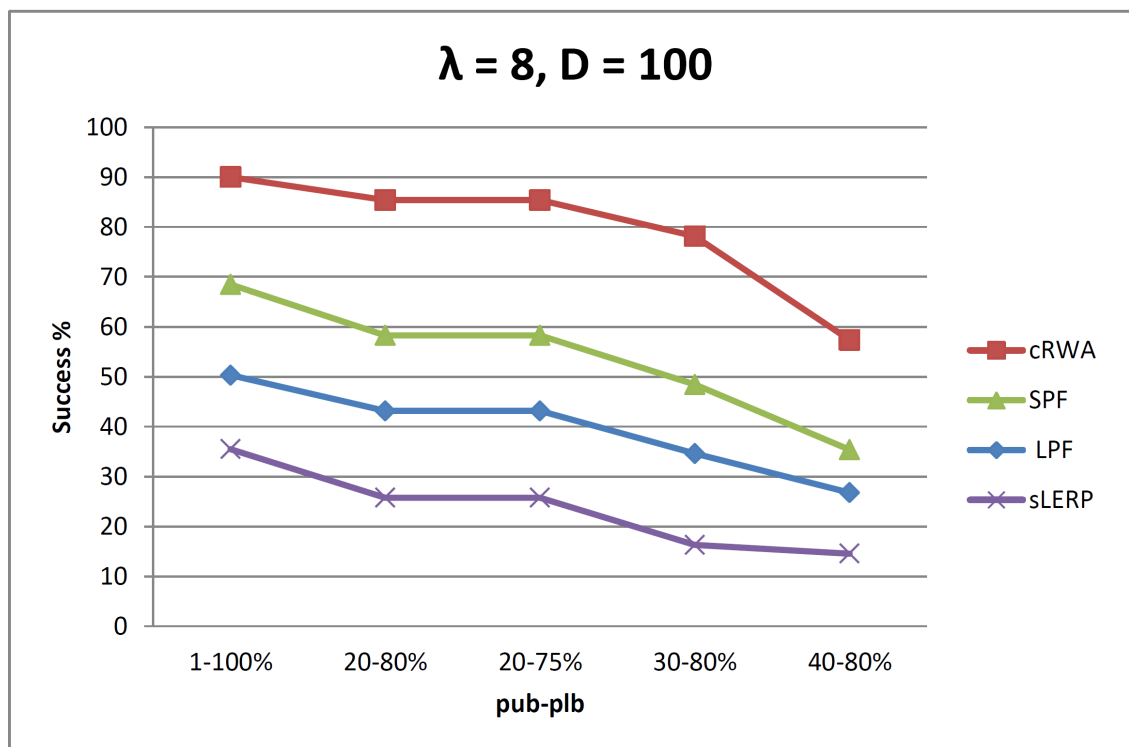


Figure 4.12: Effect of varying PERCENTAGE\_UPPER\_BOUND (*pub*) and PERCENTAGE\_LOWER\_BOUND (*plb*) in AT for  $\lambda = 8$

The results obtained are summarized in Figure 4.12 and Figure 4.13 for the values of  $\lambda = 8$  and  $\lambda = 16$ , respectively. It can be observed from both these figures that, as the difference between *plb* and *pub* gets smaller, the success rate keeps on dropping. If the range 20-80% against the range 30-80% is considered. The former range has shorter paths than the latter, hence decreasing the probability that a length of route between any source and destination will exceed the optical reach. Figure 4.14 presents the average performance of the heuristics under variable ranges.

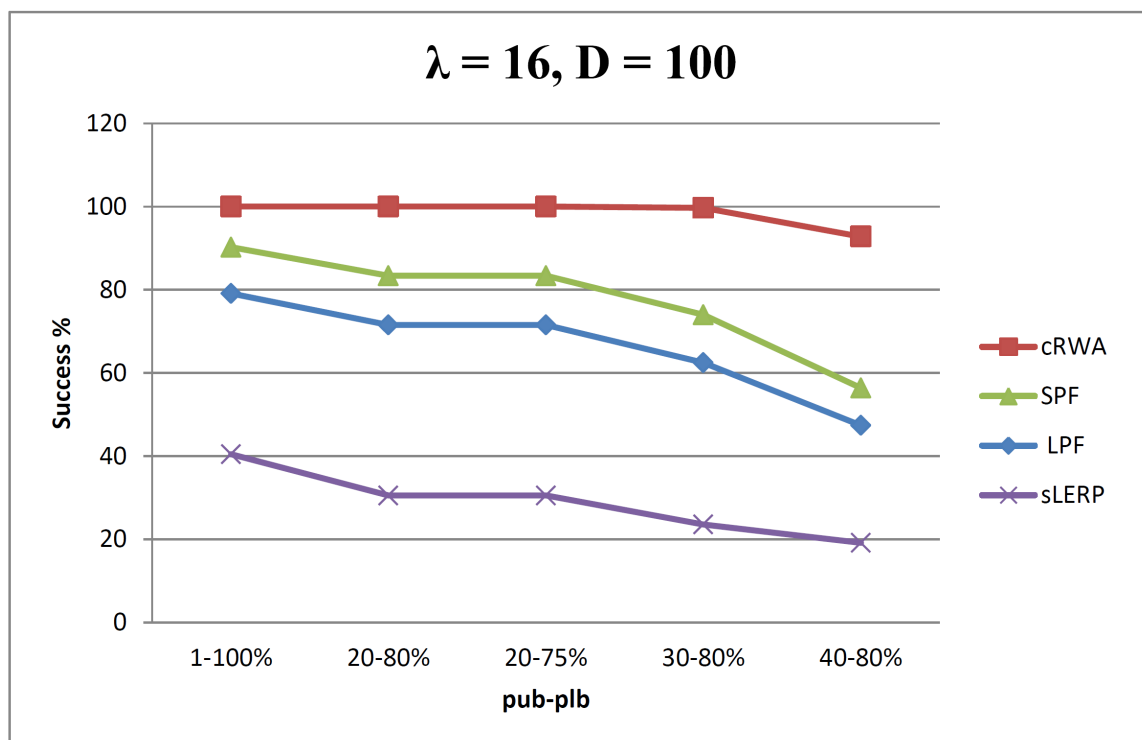


Figure 4.13: Effect of varying PERCENTAGE\_UPPER\_BOUND (*pub*) and PERCENTAGE\_LOWER\_BOUND (*plb*) in AT for  $\lambda = 16$

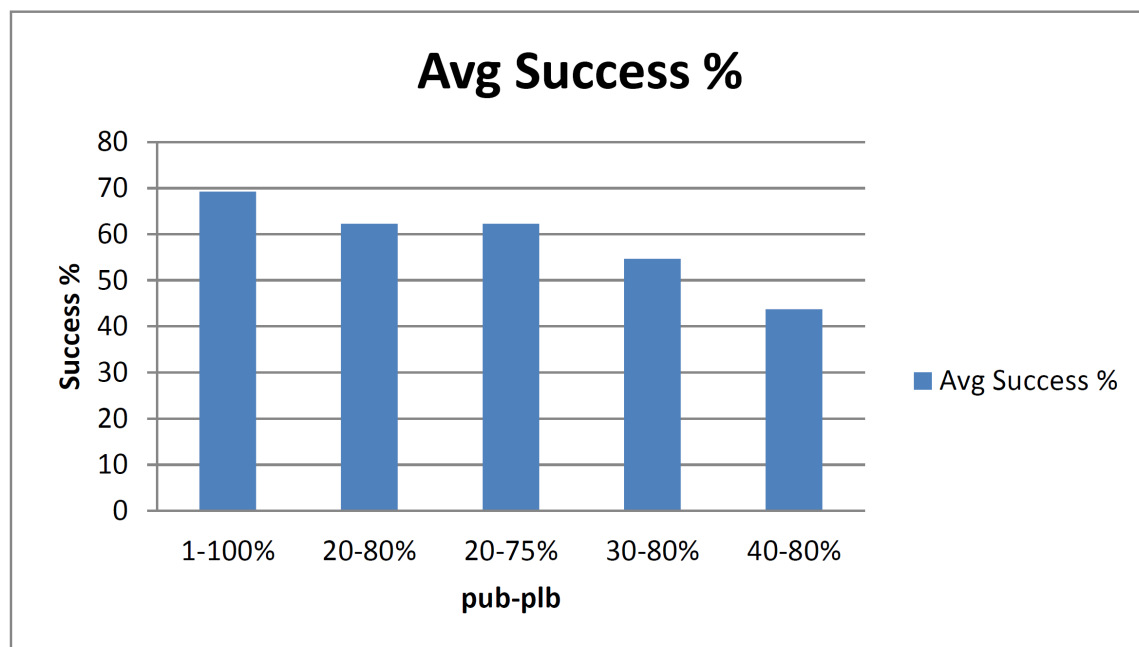


Figure 4.14: Effect of varying PERCENTAGE\_UPPER\_BOUND (*pub*) and PERCENTAGE\_LOWER\_BOUND (*plb*) in AT for  $\lambda = 16$

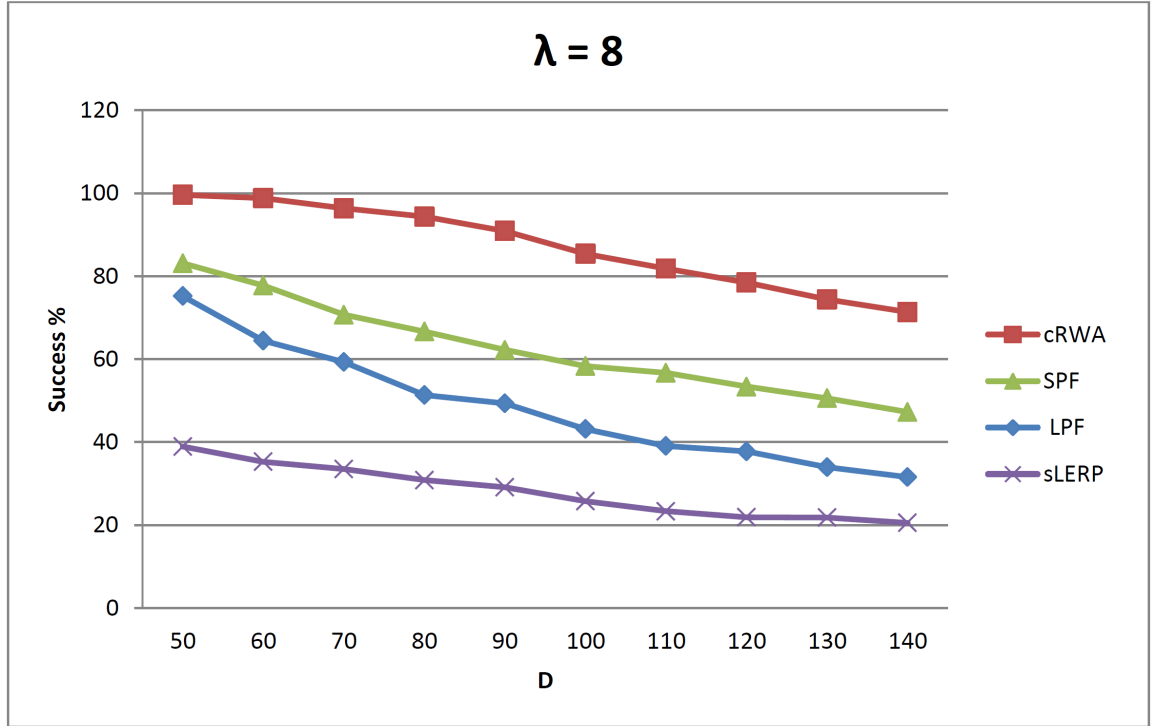


Figure 4.15: Effect of Varying  $D$  in AT

#### 4.3.1.3 Demands ( $D$ )

The effect of varying the value of  $D$  was also observed for AT. Figure 4.15 summarizes the results obtained. With sLERP performing the worst and cRWA giving the upper limit for all the heuristics, the results were consistent with the observations made in Section 4.2.3.

#### 4.3.1.4 Minimum Number ( $K$ ) of Shortest Paths between a Source-Destination Pair

As studied in Section 4.2.5, the synthetic networks did not give any consistent information when value of the parameter  $K$  was varied. This happened due to the random nature of the network and the datasets. In the case of real-life networks, the network topology is fixed and it is possible to explain the outcome of the experiments.

The parameter  $K$  guarantees that there will be at least  $K$  shortest paths between a source-destination pair in the demand set. Therefore, as the value of  $K$  increases, it improves the possibility of getting an alternate path between a source and a destination if a path is rejected (or blocked) due to some network layer. This however does not remain effective as soon as the impairments are taken into account, as the alternate paths may prove to be more prone to PLIs. In Figure 4.16, it can be observed that there are no significant changes in the

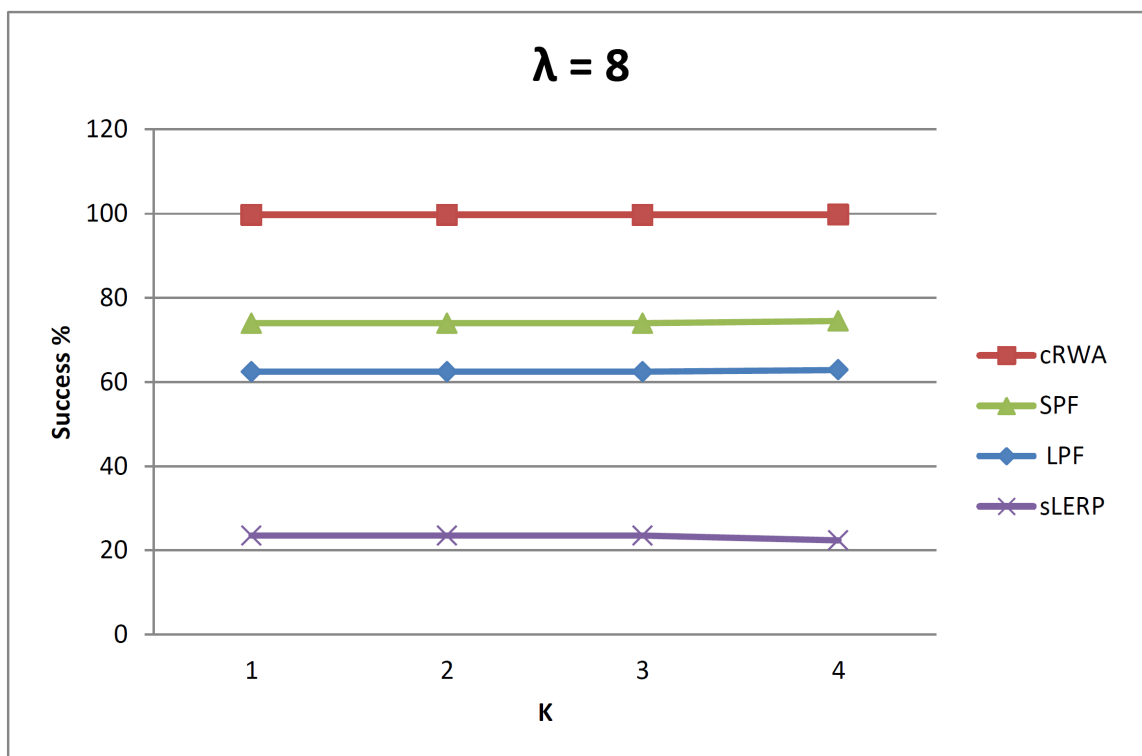


Figure 4.16: Effect of Varying  $D$  in AT

performance of the heuristics (except SPF) as the value of  $K$  is increased. In the case of SPF, a 1% gain in the performance was observed when  $K=4$ .

### 4.3.2 USANET

Figure 4.17 shows the USANET Topology obtained from [66].

#### 4.3.2.1 Channels per Fiber ( $\lambda$ )

Due to larger edge lengths, it is found that more routes exceed the optical reach. This results in smaller demand sets (in terms of the number of demands per demand set). Hence, in Figure 4.18 the behavior of the heuristics for  $\lambda=8$  and  $\lambda=16$  were observed to be identical for cRWA and sLERP. In the case of SPF and LPF, the network performed better when  $\lambda=16$ .

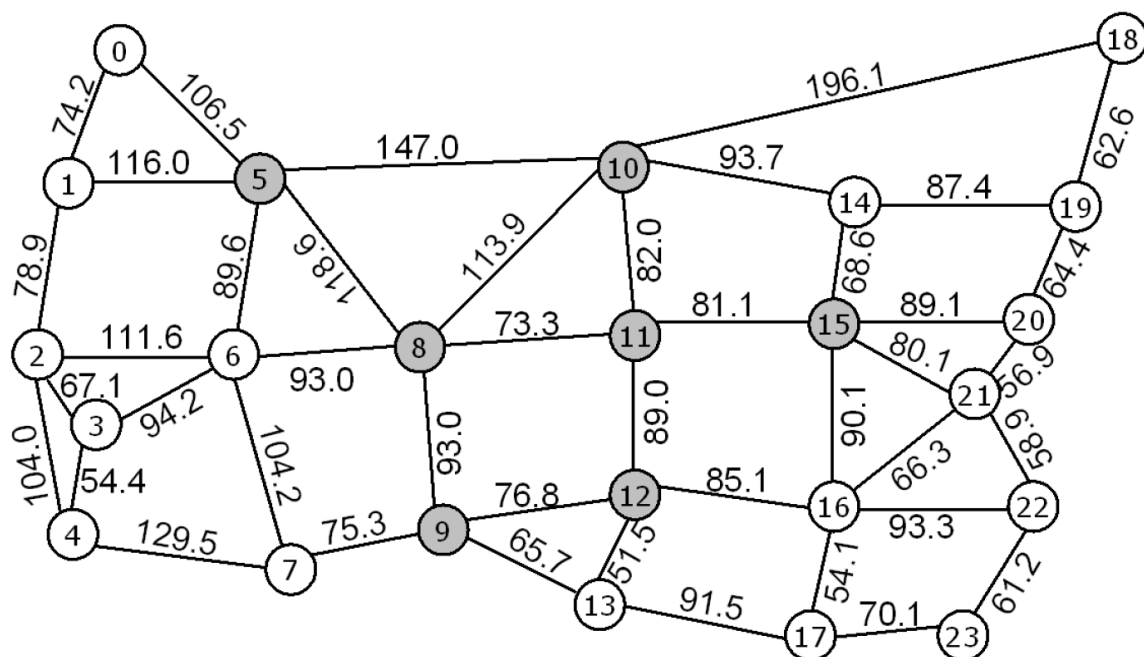
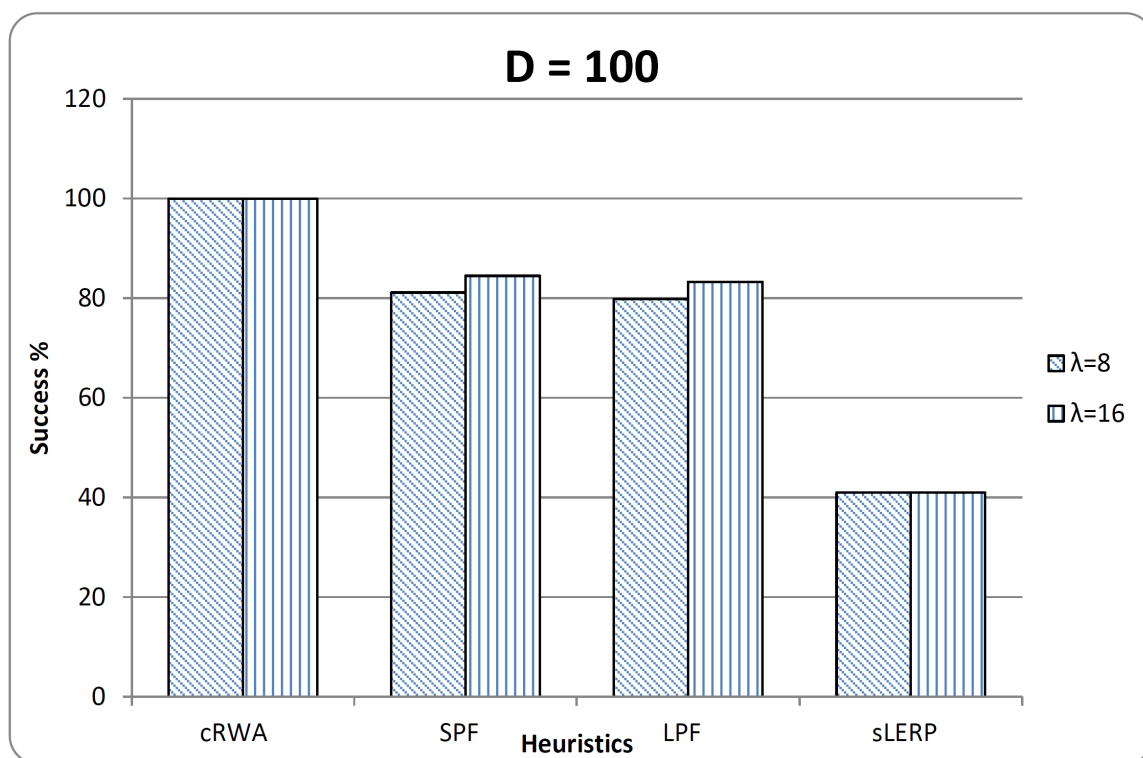


Figure 4.17: USANET

Figure 4.18: Effect of varying  $\lambda$  in USANET

#### 4.3.2.2 Demands ( $D$ )

As indicated before, the number of demands in a demand set is restricted to a smaller size, despite relaxed constraints. To ensure that the constraints have a

low effect in RDG, the range  $[plb, pub]$  and  $K$  were restricted to  $[1, 100]$  and 1, respectively. With the above relaxed constraints, the value of  $D$  was varied within the range of 50 to 140. It can be seen from Figure 4.19, that cRWA performed the best, with 100 % of the demands established in all types of demands sets. SPF and LPF were observed to have identical behavior.

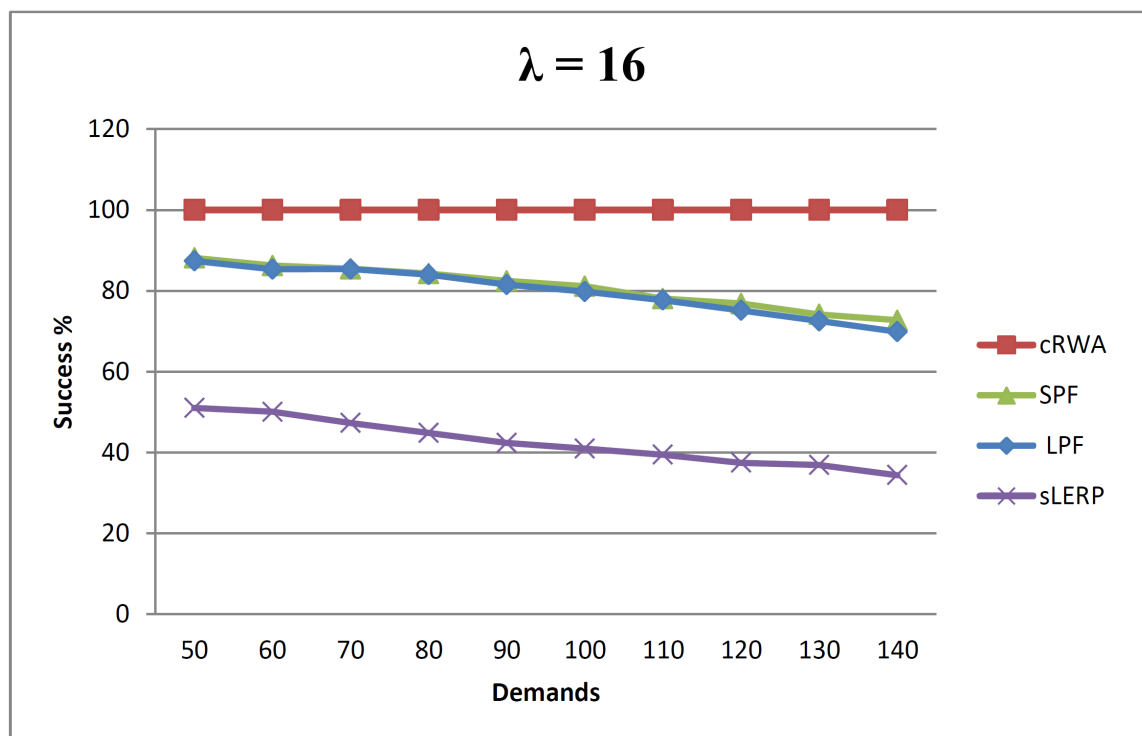


Figure 4.19: Effect of varying  $D$  USANET

On average it was observed that as the congestion in the network increases, the average success rate decreased (Figure 4.20).



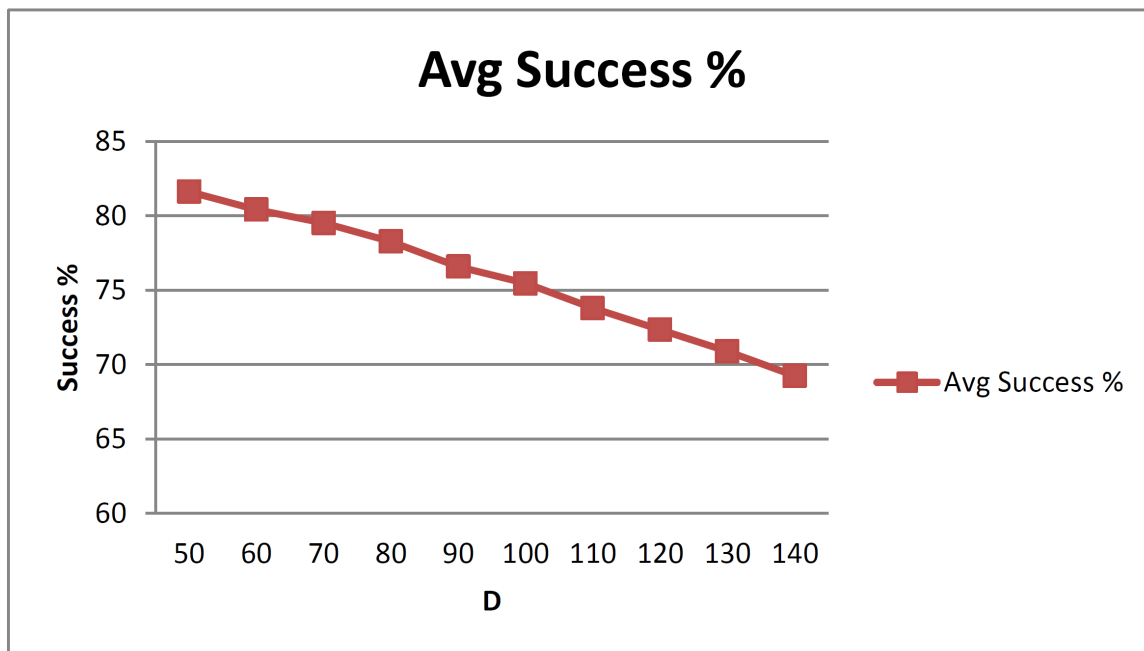


Figure 4.20: Average Success rate for varying  $D$  in USANET

#### 4.3.2.3 Minimum Number ( $K$ ) of Shortest Paths between a Source-Destination Pair

Figure 4.21 shows the effect of varying  $K$  when the heuristics are employed over USANET topology. As the value of  $K$  increases the possibility of finding an alternative path increases, hence a better performance is seen at the values  $K=2, 3$ . It can also be observed that SPF and LPF have the same identical behavior.

#### 4.3.3 ARPANET

Figure 4.22 shows the ARPANET Topology obtained from [66].

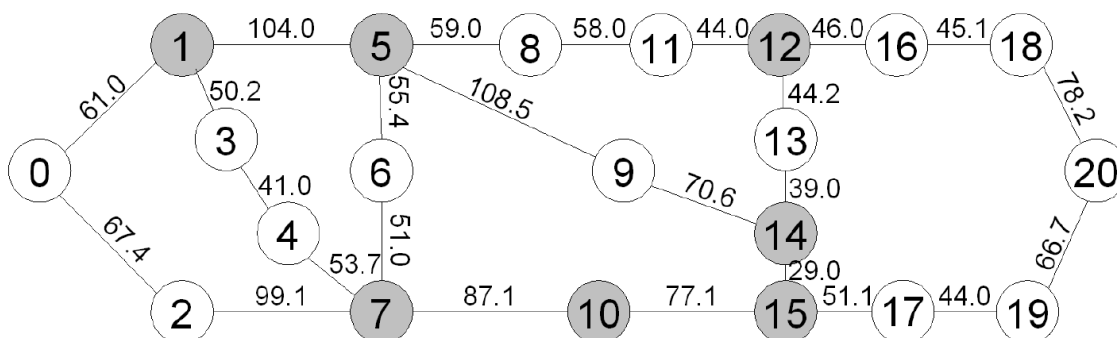


Figure 4.22: ARPANET

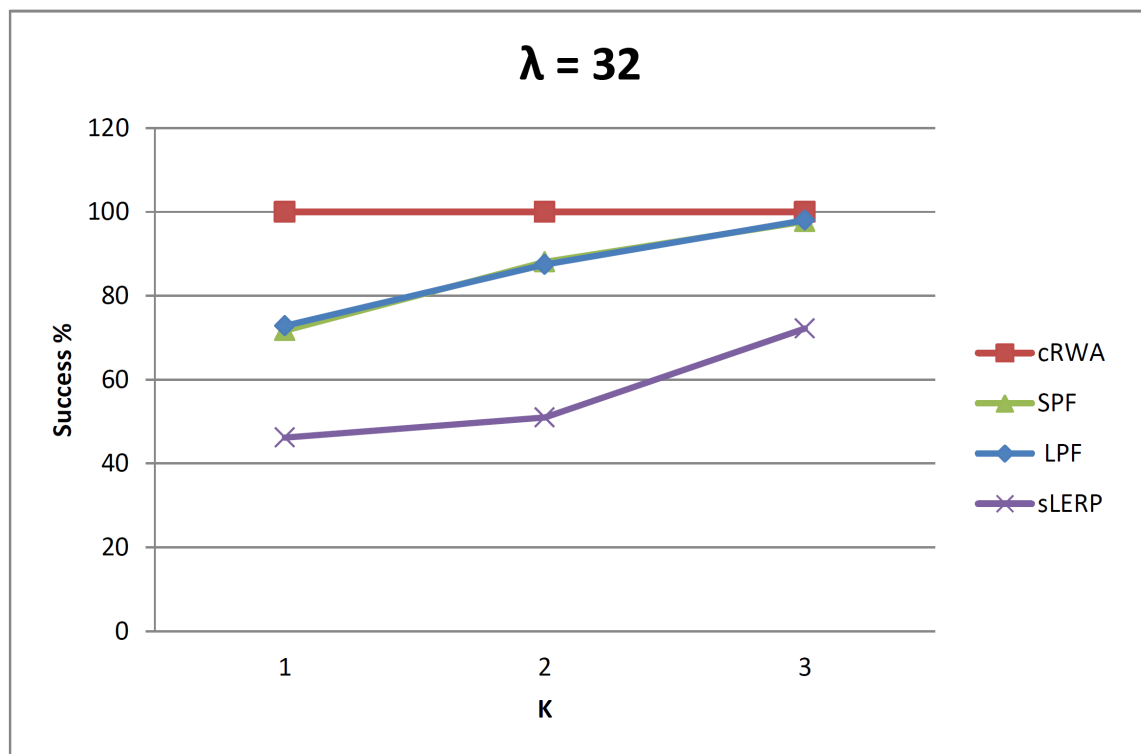
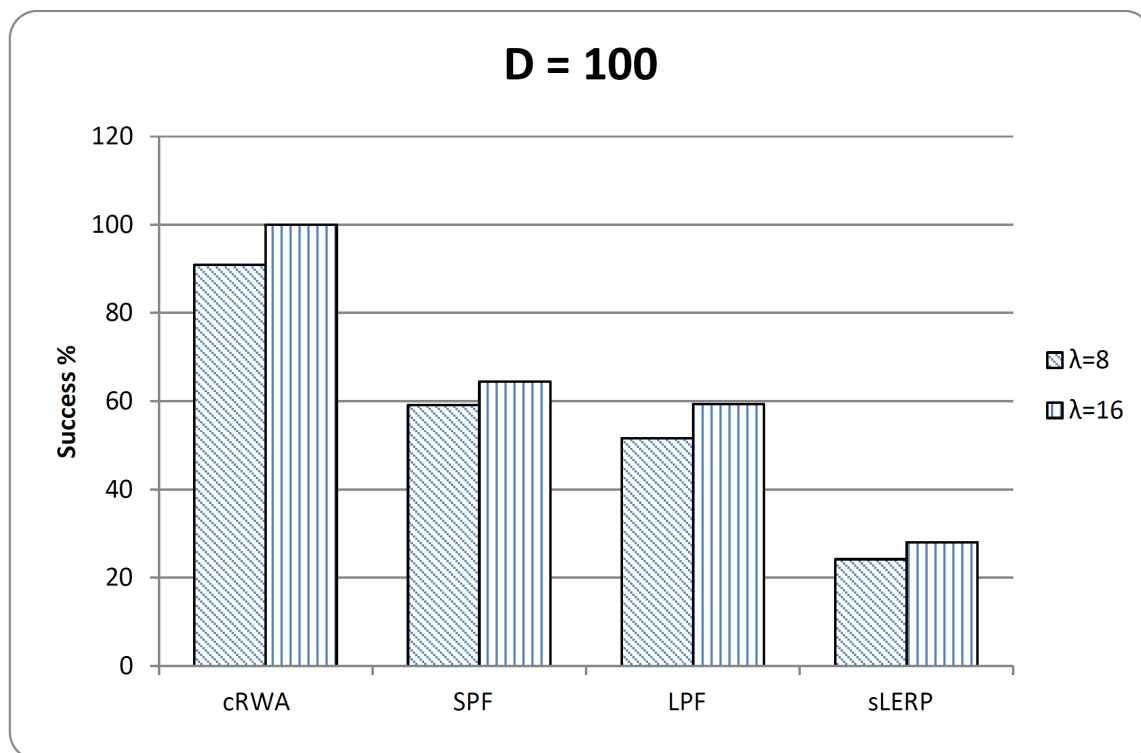


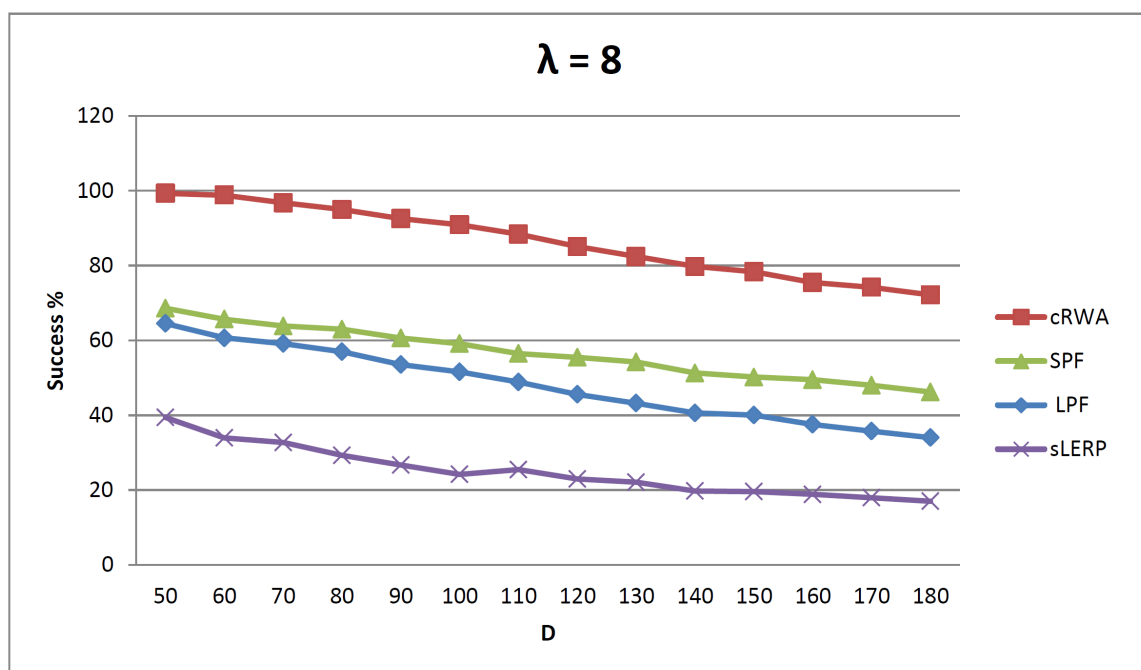
Figure 4.21: Effect of varying  $K$  values in USANET

#### 4.3.3.1 Channels per Fiber ( $\lambda$ )

Unlike USANET, the edge distances in ARPANET are shorter and hence the test-bed is able to generate larger demand sets. Figure 4.23 summarizes the results. As expected, when the number of channels per fiber ( $\lambda$ ) is 16, the performance of all the heuristics are comparatively better.

Figure 4.23: Effect of varying  $\lambda$  in ARPANET

#### 4.3.3.2 Demands ( $D$ )

Figure 4.24: Effect of varying  $D$  ARPANET

Due to relatively shorter edge lengths, the demand sets were larger in the case of ARPANET. For the experiment the following configuration was employed:

$K = 10, \lambda = 8$  and the range  $[plb, pub] = [20\%, 80\%]$ . Similar to other network topology, the performance for all the heuristics deteriorated with increase in the number of demands (Figure 4.24).

## Chapter 5

# CONCLUSIONS AND FUTURE WORK

### 5.1 Conclusions

The objective of this thesis was to develop a standard test-bed for Impairment Aware Routing and Wavelength Assignment (IA-RWA) in transparent networks. The objective of the test-bed was to compare any number of heuristics for IA-RWA in transparent optical networks based on a common criteria. With the help of some latest tools in the market like Rational Rhapsody (in design phase), Eclipse CDT – A C/C++ Developing Tool and Valgrind (in development phase), the implementation of the test-bed was successfully achieved. The test-bed was able to successfully integrate various components for the proper functioning of the heuristics. To calculate the extent of impairments, an OSNR-based PLI Tool [67] was plugged into the test-bed, which was then used by the heuristics for performance evaluation. The test-bed also includes a random network generator and a random data-set generator to supply proper inputs to the heuristics. It is observed that a  $k$ -SP algorithm is another tool that is extensively used by most of the heuristics. Hence, a  $k$ -SP algorithm [34] was also implemented as a part of the test-bed.

To test the working of the test-bed, four heuristics were implemented in this thesis. The heuristics were successfully compared against the each other. While comparing the heuristics, various parameters were tweaked in the test-bed. This type of scrutinized testing of the heuristics helps in achieving a better analysis of the heuristics. To allow performance evaluation of the heuristics on real-life networks, with some tweaks in the configuration, the test-bed can supply the heuristic with the requested type of topology. The Optical Research Group of University of Windsor was able to successfully use this tool to study a newly proposed heuristic [68].

## 5.2 Future Work

Following the guidelines of iterative software development [63], this thesis has presented the core module of the test-bed. The thesis employed the “model-view” separation approach presented by the authors of [63]. This separation gives more flexibility in choosing and prioritizing the modules. In the thesis, model was given the priority, hence very small amount of time was spent on the view part. As one of the future works, introduction of GUI (Graphical User Interface) can enhance the user experience. The user can be allowed to give inputs and tweak all kinds of parameters through the GUI. Another feature that holds the prospect in future work is the automation of the whole process with minimum user interference. The present tool needs to be manually fed with the location of the heuristic functions to run the algorithms. Using some existing technologies it is possible to read the files and execute the symbols. By specifying certain protocols for the formats of inputs and outputs, such automation can be easily achieved.

## REFERENCES

- [1] P. Moulton and J. Moulton, *Telecommunications Survival Guide*. Prentice Hall, 2001. [Online]. Available: <http://books.google.ca/books?id=qxhZAAAAYAAJ>
- [2] J. Zheng and H. Mouftah, *Optical WDM Networks: Concepts and Design Principles*. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2004, ch. 8, pp. 251 –263.
- [3] G. P. Agrawal, *Fiber-Optic Communication Systems*, 3rd ed. University of Rochester Rochester: NY: John Wiley & Sons, 2002.
- [4] S. Bandyopadhyay, *Dissemination of Information in Optical Networks: From Technology to Algorithm*, W. Brauer and J. Hromkovic, Eds. University of Windsor 401 Sunset Avenue Windsor, Ontario N9B 3P4: Springer Publications, 2008.
- [5] Fabila. (2008, January) Principles of the transmission by optical fiber. website. Jesus Infiesta Saborit. [Online]. Available: <http://www.fabila.com/noticia.asp?id=683>
- [6] T. Networks, “Fiber optics basics,” <http://www.transition.com/TransitionNetworks/>, 6475 City West Parkway Minneapolis, MN 55344 USA, 2012. [Online]. Available: [http://www.transition.com/transitionnetworks/Resources/en/PDF/fiber\\_wp.pdf](http://www.transition.com/transitionnetworks/Resources/en/PDF/fiber_wp.pdf)
- [7] Wikipedia, “Total internal reflection — wikipedia, the free encyclopedia,” 2012, [Online; accessed 12-May-2012]. [Online]. Available: [http://en.wikipedia.org/w/index.php?title=Total\\_internal\\_reflection&oldid=489030158](http://en.wikipedia.org/w/index.php?title=Total_internal_reflection&oldid=489030158)
- [8] G. Papadimitriou, C. Papazoglou, and A. Pomportsis, “Optical switching: switch fabrics, techniques, and architectures,” *Journal of Lightwave Technology*, vol. 21, no. 2, pp. 384 – 405, Feb. 2003.

- [9] M. R. Ang Ngang Lim, "Switching fabric," September 2005. [Online]. Available: <http://searchstorage.techtargget.com/definition/switching-fabric>
- [10] R. Ramaswami, K. N. Sivarajan, and G. Hajime Sasaki, *Optical Networks: A Practical Perspective*. Morgan Kaufmann, 2010.
- [11] S. Kartalopoulos, *Optical Bit Error Rate: An Estimation Methodology*, 1st ed. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2004, ch. 3, pp. 81 –100.
- [12] K. S., *Optical Bit Error Rate: An Estimation Methodology*, 1st ed. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2004, ch. 1, pp. 7–25.
- [13] G. Keiser, *FTTX Concepts and Applications*, 1st ed. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2006, ch. 4, pp. 61 – 78.
- [14] Wikipedia, "Attenuation — wikipedia the free encyclopedia," 2012, [Online; accessed 11-May-2012]. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=Attenuation&oldid=489007978>
- [15] H. Majedi, "Signal degradation in optical fibers," Ph.D. dissertation, January 2006.
- [16] S. Kartalopoulos, *DWDM Networks, Devices, and Technology*, 1st ed. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2002, ch. 2, pp. 93 –234.
- [17] S. Azodolmolky, M. Klinkowski, E. Marin, D. Careglio, J. S. Pareta, and I. Tomkos, "A survey on physical layer impairments aware routing and wavelength assignment algorithms in optical networks," *Computer Networks*, vol. 53, no. 7, pp. 926 – 944, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128608003964>
- [18] T. Numai, *Laser Diodes and Their Applications to Communications and Information Processing*, 1st ed. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2011, ch. 9, pp. 233 –245.
- [19] S. Kartalopoulos, *Introduction to DWDM: Technology: Data in a Rainbow*. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2000, ch. 4, pp. 177–177.



- [20] Fiber-Optics.Info. (2012) Dense wavelength-division multiplexing. <http://www.fiber-optics.info/articles/>. Fiber-Optics.Info. [Online]. Available: [http://www.fiber-optics.info/articles/dense\\_wavelength-division\\_multiplexing](http://www.fiber-optics.info/articles/dense_wavelength-division_multiplexing)
- [21] J. M. McQuillan, "Graph theory applied to optimal connectivity in computer networks," *SIGCOMM Comput. Commun. Rev.*, vol. 7, no. 2, pp. 13–41, Apr. 1977. [Online]. Available: <http://doi.acm.org/10.1145/1024857.1024860>
- [22] H. Zang and J. P. Jue, "A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks," *Optical Networks Magazine*, vol. 1, pp. 47–60, 2000.
- [23] G. N. Rouskas, "Routing and wavelength assignment in optical wdm networks," 2001.
- [24] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *16th Annual Symposium on Foundations of Computer Science, 1975.*, Oct 1975, pp. 184–193.
- [25] M. Gamst, *A Survey of the Routing and Wavelength Assignment Problem*. DTU Management, 2009.
- [26] Optical-Hub. (2010, July) "terminology list". <http://www.optical-network.com/terminology.php>. Optical Hub. [Online]. Available: <http://www.optical-network.com/terminology.php??letter=all&id=36>
- [27] B. Ramamurthy, H. Feng, D. Datta, J. Heritage, and B. Mukherjee, "Transparent vs. opaque vs. translucent wavelength-routed optical networks," in *Optical Fiber Communication Conference, and the International Conference on Integrated Optics and Optical Fiber Communication. OFC/IOOC '99. Technical Digest*, vol. 1, 1999, pp. 59–61.
- [28] S.-W. Seo, K. Bergman, and P. Prucnal, "Transparent optical networks with time-division multiplexing," *IEEE Journal on Selected Areas in Communications.*, vol. 14, no. 5, pp. 1039–1051, jun 1996.
- [29] J. Nellist, *Understanding Telecommunications and Lightwave Systems: An Entry Level Guide*. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2002, ch. 11, pp. 78–91.
- [30] C. Saradhi and S. Subramaniam, "Physical layer impairment aware routing (pliar) in wdm optical networks: issues and challenges," *Communications Surveys Tutorials, IEEE*, vol. 11, no. 4, pp. 109–130, quarter 2009.

- [31] M. Ezzahdi, S. Al Zahr, M. Koubaa, N. Puech, and M. Gagnaire, “Lerp: a quality of transmission dependent heuristic for routing and wavelength assignment in hybrid wdm networks,” in *Proceedings 15th International Conference on Computer Communications and Networks, 2006. ICCCN 2006.*, Oct. 2006, pp. 125–136.
- [32] P. Tomsu and C. Schmutzer, *Next Generation Optical Networks: The Convergence of IP Intelligence and Optical Technologies*, D. Cullen-Dolce, Ed. Prentice Hall PTR Upper Saddle River NJ 07458: Prentice Hall PTR, 2002.
- [33] K. Thyagarajan and A. . Ghatak, *Fiber Optic Essentials*. IEEE Press 445 Hoes Lane Piscataway, NJ 08854: Wiley-IEEE Press, 2007, ch. 13, pp. 205–220.
- [34] J. Y. Yen, “Finding the k-shortest loopless paths in a network,” *Management Science*, vol. 17, no. 11, pp. 712–716, 1971. [Online]. Available: <http://www.jstor.org/stable/2629312>
- [35] E. de Queiros Vieira Martins, “An algorithm for ranking paths that may contain cycles,” *European Journal of Operational Research*, vol. 18, no. 1, pp. 123 – 130, 1984. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0377221784902698>
- [36] J. A. Azevedo, M. E. O. S. Costa, J. J. E. S. Madeira, and E. Q. V. Martins, “An algorithm for the ranking of shortest paths,” *European Journal of Operational Research*, vol. 69, no. 1, pp. 97 – 106, 1993. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0377221793900955>
- [37] J. Y. Yen, *Finding All Shortest Paths from a Fixed Node in N-node Non-negative-distance Complete Networks Using 1/2N<sup>2</sup> Comparisons*, ser. Working paper (University of Santa Clara. Graduate School of Business). University of Santa Clara, 1972. [Online]. Available: <http://books.google.ca/books?id=JfqwPgAACAAJ>
- [38] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959, 10.1007/BF01386390. [Online]. Available: <http://dx.doi.org/10.1007/BF01386390>
- [39] I. Tomkos, D. Vogiatzis, C. Mas, I. Zacharopoulos, A. Tzanakaki, and E. Varvarigos, “Performance engineering of metropolitan area optical networks through impairment constraint routing,” *Communications Magazine, IEEE*, vol. 42, no. 8, pp. S40 – S47, Aug. 2004.

- [40] K. Manousakis, K. Christodoulopoulos, and E. Varvarigos, "Impairment-aware offline rwa for transparent optical networks," in *INFOCOM 2009, IEEE*, April 2009, pp. 1557–1565.
- [41] K. Christodoulopoulos, K. Manousakis, and E. Varvarigos, "Offline routing and wavelength assignment in transparent wdm networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1557–1570, 2010.
- [42] P. Pavon-Marino, S. Azodolmolky, R. Aparicio-Pardo, B. Garcia-Manrubia, Y. Pointurier, M. Angelou, J. Sole-Pareta, J. Garcia-Haro, and I. Tomkos, "Offline impairment aware rwa algorithms for cross-layer planning of optical networks," *Journal of Lightwave Technology*, vol. 27, no. 12, pp. 1763–1775, june15, 2009.
- [43] S. Zsigmond, G. Németh, and T. Cinkler, "Mutual impact of physical impairments and grooming in multilayer networks," in *Proceedings of the 11th International IFIP TC6 Conference on Optical Network Design and Modeling*, ser. ONDM'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 38–47. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1768792.1768797>
- [44] J. He, M. Brandt-Pearce, Y. Pointurier, and S. Subramaniam, "Qot-aware routing in impairment-constrained optical networks," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, Nov. 2007, pp. 2269–2274.
- [45] D. A. Chaves, D. O. Aguiar, C. J. Bastos-Filho, and J. F. Martins-Filho, "Fast and adaptive impairment aware routing and wavelength assignment algorithm optimized by offline simulations," *Optical Switching and Networking*, vol. 7, no. 3, pp. 127–138, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1573427710000184>
- [46] R. Cardillo, V. Curri, and M. Mellia, "Considering transmission impairments in configuring wavelength routed optical networks," in *Optical Fiber Communication Conference, 2006 and the 2006 National Fiber Optic Engineers Conference. OFC 2006*, 2006, p. 3 pp.
- [47] Y. Huang, J. Heritage, and B. Mukherjee, "Connection provisioning with transmission impairment consideration in optical wdm networks with high-speed channels," *Journal of Lightwave Technology*, vol. 23, no. 3, pp. 982–993, march 2005.
- [48] B. Ramamurthy, D. Datta, H. Feng, J. Heritage, and B. Mukherjee, "Impact of transmission impairments on the teletraffic performance of wavelength-

- routed optical networks,” *Journal of Lightwave Technology*, vol. 17, no. 10, pp. 1713–1723, 1999.
- [49] H. Pereira, D. Chaves, C. Bastos-Filho, and J. Martins-Filho, “Impact of physical layer impairments in all-optical networks,” in *Microwave and Optoelectronics Conference, 2007. IMOC 2007. SBMO/IEEE MTT-S International*, 29 2007–Nov. 2007, pp. 536–541.
- [50] J. He, M. Brandt-Pearce, Y. Pointurier, C. Brown, and S. Subramaniam, “Adaptive wavelength assignment using wavelength spectrum separation for distributed optical networks,” in *IEEE International Conference on Communications, 2007. ICC '07.*, june 2007, pp. 2406–2411.
- [51] J. He, M. Brandt-Pearce, and S. Subramaniam, “Qos-aware wavelength assignment with ber and latency guarantees for crosstalk limited networks,” in *IEEE International Conference on Communications. ICC '07.*, June 2007, pp. 2336–2341.
- [52] C. Mas, D. Vogiatzis, and I. Tomkos, “Impairment aware routing in metropolitan area optical networks,” in *. Proceedings of 2004 6th International Conference on Transparent Optical Networks, 2004*, vol. 1, July 2004, pp. 91–96 vol.1.
- [53] N. Sengezer and E. Karasan, “Static lightpath establishment in multilayer traffic engineering under physical layer impairments,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 9, pp. 662–677, 2010.
- [54] G. Markidis, S. Sygletos, A. Tzanakaki, and I. Tomkos, “Impairment aware based routing and wavelength assignment in transparent long haul networks,” in *Optical Network Design and Modeling*, ser. Lecture Notes in Computer Science, I. Tomkos, F. Neri, J. SolA Pareta, X. Masip Bruin, and S. SAinchez Lopez, Eds. Springer Berlin / Heidelberg, 2007, vol. 4534, pp. 48–57. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-72731-6\\_6](http://dx.doi.org/10.1007/978-3-540-72731-6_6)
- [55] M. Bakri, M. Koubaa, M. Menif, and I. Ouerda, “Static lightpath establishment with transmission impairments consideration in wdm all-optical networks,” in *Design of Reliable Communication Networks. DRCN 2009. 7th International Workshop on*, Oct. 2009, pp. 251–258.
- [56] S. Azodolmolky, Y. Pointurier, M. Klinkowski, E. Marin, D. Careglio, J. Sole-Pareta, M. Angelou, and I. Tomkos, “On the offline physical layer impairment aware rwa algorithms in transparent optical networks: State-of-the-art

and beyond,” in *International Conference on Optical Network Design and Modeling, ONDM 2009.*, Feb. 2009, pp. 1–6.

- [57] S. Pachnicke, T. Paschenda, and P. Krummrich, “Assessment of a constraint-based routing algorithm for translucent 10gbits/s dwdm networks considering fiber nonlinearities,” *Journal of Optical Network.*, vol. 7, no. 4, pp. 365–377, Apr 2008. [Online]. Available: <http://jon.osa.org/abstract.cfm?URI=jon-7-4-365>
- [58] A. Keles, A. Yayimli, and A. Uyar, “Ant based hyper-heuristic for physical impairment aware routing and wavelength assignment,” in *Sarnoff Symposium, 2010 IEEE*, April 2010, pp. 1–5.
- [59] M. Dorigo, “Optimization, Learning and Natural Algorithms,” Ph.D. dissertation, Politecnico di Milano, Italy, 1992.
- [60] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, A.-W. Editors, Ed. Addison-Wesley, 1989. [Online]. Available: <http://www.amazon.com/Genetic-Algorithms-Optimization-Machine-Learning/dp/0201157675>
- [61] D. Monoyios and K. Vlachos, “Multiobjective genetic algorithms for solving the impairment-aware routing and wavelength assignment problem,” *Journal of Optical Communication Network.*, vol. 3, no. 1, pp. 40–47, Jan 2011. [Online]. Available: <http://jocn.osa.org/abstract.cfm?URI=jocn-3-1-40>
- [62] G. Scholar, “Citations for the paper - lerp: a quality of transmission dependent heuristic for routing and wavelength assignment in hybrid wdm networks,” 2012, [Online; accessed 17-May-2012]. [Online]. Available: [http://scholar.google.ca/scholar?as\\_sdt=2005&hl=en&sciodt=0,5&cites=13015794404384799176&scipsc=](http://scholar.google.ca/scholar?as_sdt=2005&hl=en&sciodt=0,5&cites=13015794404384799176&scipsc=)
- [63] C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Prentice Hall PTR Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004.
- [64] A. Newell and H. A. Simon, “Computer science as empirical inquiry: symbols and search,” *Commun. ACM*, vol. 19, no. 3, pp. 113–126, 1976. [Online]. Available: <http://doi.acm.org/10.1145/360018.360022>
- [65] M. D. Atkinson, J.-R. Sack, N. Santoro, and T. Strothotte, “Min-max heaps and generalized priority queues,” *Communications Magazine. ACM*, vol. 29, no. 10, pp. 996–1000, Oct 1986. [Online]. Available: <http://doi.acm.org/10.1145/6617.6621>

- [66] G. Shen, W. D. Grover, T. H. Cheng, and S. K. Bose, “Sparse placement of electronic switching nodes for low-blocking in translucent optical networks,” *Journal of Optical Networking*, vol. 1, pp. 424–441, 2002.
- [67] H. Pereira, D. Chaves, C. Bastos-Filho, and J. Martins-Filho, “Osnr model to consider physical layer impairments in transparent optical networks,” *Photonic Network Communications*, vol. 18, pp. 137–149, 2009, 10.1007/s11107-008-0178-2. [Online]. Available: <http://dx.doi.org/10.1007/s11107-008-0178-2>
- [68] S. Zawada, “Impairment-aware static route and wavelength assignment in wdm networks,” Master’s thesis, University of Windsor, 2012.

## VITA AUCTORIS

NAME Shrestharth Ghosh  
PLACE OF BIRTH: Dehradun, India  
YEAR OF BIRTH: 1987  
EDUCATION: Bachelor of Computer Science (BCS) [General]  
University of Windsor  
Windsor, Ontario, Canada, 2009-2010  
  
Bachelor of Engineering (Computer Science)  
Visvesvaraya Technological University  
Bangalore, India, 2004-2008