Master of Science Thesis

Impairment-aware Survivable Routing and Regenerator Placement in WDM Networks

Desalegn Segni Abawollo

Delft University of Technology

Delft University of Technology

Faculty of Electrical Engineering, Mathematics & Computer Science
Network Architectures and Services

Impairment-aware Survivable Routing and Regenerator Placement in WDM Networks

A thesis submitted in partial fulfillment of Master of Science in Computer Engineering

Desalegn Segni Abawollo
(1532618)

Committee Members:

Supervisor: Dr. Ir. Fernando A. Kuipers
Mentor: Ir. Anteneh A. Beshir
Others: Prof. Dr. Ir. Piet Van Mieghem
Dr. Ir. Jos H. Weber

August 31, 2010
Thesis No: PVM 2010-064

Delft University of Technology

Abstract

Optical networks employing the wavelength division multiplexing (WDM) technology are promising solutions to the ever increasing demand for bandwidth. The use of WDM allows aggregation of many channels onto a single fibre without the need of high speed optoelectronic devices for end users.

In WDM networks, as the optical signals traverse multiple links their quality deteriorates due to the physical impairments they encounter. This necessitates regeneration of the signals at the intermediate nodes so that the signals will reach the destination with an acceptable level of quality. In addition, due to the frequent occurrence of fiber cuts and the tremendous amount of data transported, survivability, which is the ability to reconfigure and reestablish communication upon failure, is indispensable in WDM networks. Survivability is of critical importance in high-speed optical communication networks.

These days, network survivability along with impairment-aware routing is of great interest to telecommunication system vendors, service providers and end users.

This thesis work focuses on solving impairment-aware survivable routing in WDM networks. We have proposed and implemented exact and heuristic algorithms that solve survivable regenerator placement and survivable impairment-aware routing problems in WDM networks.

Acknowledgements

This thesis was carried out at the Network Architectures and Services group at the faculty of Electrical Engineering, Mathematics and Computer Science at Delft University of Technology.

First I would like to express my sincere appreciation to Ir. Anteneh A. Beshir, my daily mentor. Anteneh spend a lot of time giving me advice, commenting my work and helping me with difficulties. I am truly grateful for all his help.

My deepest gratitude goes out to my supervisor Dr. Ir. Fernando Kuipers who has always taken the time to give comments for improvement of my work. His door was always open even though he has a busy schedule.

During my time in Delft many people gave me their suggestions and help in my study. I would like to express my acknowledgment to them.

Last but not least, I would like to express my thanks to my family, who have always supported me with the choices I have made.

Delft University of Technology — Desalegn S. Abawollo
August 2010

Table of Contents

Abstract **iii**

Acknowledgements **v**

Acronyms **xiii**

1 Introduction **1**

1-1 Basic Concepts in WDM Optical networks 1

1-1-1 Routing and Wavelength Assignment (RWA) 2

1-1-2 Physical Layer Impairments 3

1-1-3 WDM Optical Network Survivability 5

1-2 Related Work 10

1-3 Objectives 12

1-4 Organization of the Thesis 12

2 Survivable Regenerator Placement Problem **15**

2-1 Dedicated Survivable Regenerator Placement Problem 16

2-1-1 Problem Definition 16

2-1-2 Exact Algorithm 16

2-1-3 Single Physical Impairment Heuristic Algorithm 20

2-1-4 Simulation Results 21

2-2 Shared Survivable Regenerator Placement Problem 25

2-2-1 Problem Definition 25

2-2-2 Exact Algorithm 25

2-2-3 Single Physical Impairment Heuristic Algorithm 26

2-2-4 Simulation Results . . . 28
2-3 Backup Shared Survivable Regenerator Placement Problem . . . 29
2-3-1 Problem Definition . . . 29
2-3-2 Single Physical Impairment Heuristic Algorithm . . . 30
2-3-3 Simulation Results . . . 31
2-4 Multiple Physical Impairments Heuristic Algorithms . . . 34
2-4-1 Dedicated Survivable Regenerator Placement . . . 34
2-4-2 Shared Survivable Regenerator Placement . . . 35
2-4-3 Backup Shared Survivable Regenerator Placement . . . 36
2-4-4 Simulation Results . . . 37

3 Impairment-aware Survivable Routing Problem 43
3-1 Dedicated Impairment-aware Survivable Routing Problem . . . 43
3-1-1 Problem Definition . . . 45
3-1-2 Exact Algorithm . . . 45
3-1-3 Heuristic Algorithms . . . 50
3-1-4 Simulation Results . . . 54
3-2 Shared Impairment-aware Survivable Routing Problem . . . 61
3-2-1 Problem Definition . . . 61
3-2-2 Simulation Results . . . 62
3-3 Backup Shared Impairment-aware Survivable Routing Problem . . . 67
3-3-1 Problem Definition . . . 67
3-3-2 Simulation Results . . . 68
3-4 Impairment-aware Survivable Routing Problem under dynamic traffic . . . 73
3-4-1 Unlimited wavelength . . . 73
3-4-2 Limited wavelength per link . . . 76
3-5 Backup WR Shared Impairment-aware Survivable Routing Problem . . . 77
3-5-1 Simulation Results . . . 77

4 Discussion, Conclusion and Future Work 81
4-1 Discussion . . . 81
4-2 Conclusion . . . 82
4-3 Future Work . . . 83

A Additional Results 91
A-1 Survivable Regenerator Placement Problem . . . 91
A-1-1 Multiple Physical Impairments Heuristic Algorithms . . . 91
A-2 Impairment-aware Survivable Routing Problem . . . 96

List of Figures

2-1 The NSFNET network. 22
2-2 The ARPANET network. 22
2-3 Average regenerator number vs physical impairment threshold for lattice network with $N = 25$ (Exact vs DRPB). 23
2-4 Average regenerator number vs physical impairment threshold for random network with $N = 25$ (Exact vs DRPB). 23
2-5 Average regenerator number vs physical impairment threshold for the ARPANET network (Exact vs DRPB). 24
2-6 Average regenerator number vs physical impairment threshold for the ARPANET network (Exact_S vs SRPB). 28
2-7 Average regenerator number vs physical impairment threshold for the ARPANET network (DRPB vs SRBB). 32
2-8 Average regenerator number vs physical impairment threshold for lattice network with $N = 49$ (DRPB vs SRBB). 32
2-9 Average regenerator number vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (DRPB vs SRBB). 33
2-10 Average regenerator number vs physical impairment threshold for NSFNET network (MPI_Exact vs MPI_DRPB). 38
2-11 Average regenerator number vs physical impairment threshold for NSFNET network (MPI_Exact_S vs MPI_SRPB). 39
2-12 Running time vs physical impairment threshold for NSFNET network (MPI_Exact vs MPI_DRPB). 39
2-13 Running time vs physical impairment threshold for NSFNET network (MPI_Exact_S vs MPI_SRPB). 40
2-14 Average regenerator number vs physical impairment threshold for the ARPANET network (MPI_DRPB vs MPI_SRPB). 40

2-15 Average regenerator number vs physical impairment threshold for the ARPANET network (MPI_DRPB vs MPI_SRBB). 41
2-16 Running time vs physical impairment threshold for the ARPANET network (MPI_DRPB vs MPI_SRPB). 42
2-17 Running time vs physical impairment threshold for the ARPANET network (MPI_DRPB vs MPI_SRBB). 42

3-1 An example showing that the dominance concept fails for impairment-aware routing. 46
3-2 Acceptance ratio vs physical impairment threshold for the ARPANET network with $N_R = 12$. 54
3-3 Average regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 12$. 55
3-4 Running time vs physical impairment threshold for the ARPANET network with $N_R = 12$. 56
3-5 Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 58
3-6 Average regenerator number vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 58
3-7 Running time vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 59
3-8 Acceptance Ratio vs physical impairment threshold for the ARPANET network with $N_R = 12$ (shared regenerator assignment between primary and backup path). 62
3-9 Average regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 12$ (shared regenerator assignment between primary and backup path). 63
3-10 Running time vs physical impairment threshold for the ARPANET network with $N_R = 12$ (shared regenerator assignment between primary and backup path). . 64
3-11 Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1 (shared regenerator assignment between primary and backup path). 64
3-12 Average regenerator number vs total number of regenerators for the ARPANET network with physical impairment threshold = 1 (shared regenerator assignment between primary and backup path). 65
3-13 Running time vs total number of regenerators for the ARPANET network with physical impairment threshold = 1 (shared regenerator assignment between primary and backup path). 66
3-14 Acceptance Ratio vs physical impairment threshold for the ARPANET network with $N_R = 20$. 68
3-15 Path total regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 20$. 69
3-16 Running time vs physical impairment threshold for the ARPANET network with $N_R = 20$. 70

3-17 Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 71
3-18 Path total regenerator number vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 71
3-19 Running time vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 72
3-20 Acceptance Ratio vs physical impairment threshold for the ARPANET network with $N_R = 20$. 74
3-21 Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 75
3-22 Acceptance Ratio vs physical impairment threshold for the ARPANET network with $N_R = 20$. 76
3-23 Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 77
3-24 Acceptance ratio vs physical impairment threshold for the ARPANET network with $N_R = 20$. 78
3-25 Path total regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 20$. 79
3-26 Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 79
3-27 Path total regenerator number vs total number of regenerators for the ARPANET network with physical impairment threshold = 1. 80

A-1 Average regenerator number vs physical impairment threshold for lattice network with $N = 49$ (MPI_DRPB vs MPI_SRPB). 91
A-2 Average regenerator number vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (MPI_DRPB vs MPI_SRPB). 92
A-3 Average regenerator number vs physical impairment threshold for lattice network with $N = 49$ (MPI_DRPB vs MPI_SRBB). 92
A-4 Average regenerator number vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (MPI_DRPB vs MPI_SRBB). 93
A-5 Running time vs physical impairment threshold for lattice network with $N = 49$ (MPI_DRPB vs MPI_SRPB). 93
A-6 Running time vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (MPI_DRPB vs MPI_SRPB). 94
A-7 Running time vs physical impairment threshold for lattice network with $N = 49$ (MPI_DRPB vs MPI_SRBB). 94
A-8 Running time vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (MPI_DRPB vs MPI_SRBB). 95
A-9 Acceptance ratio vs physical impairment threshold for NSFNET network ($N_R = 6$). 96
A-10 Acceptance ratio vs physical impairment threshold for lattice network with $N = 49$ and $N_R = 12$. 96

A-11 Average regenerator number vs physical impairment threshold for NSFNET network with $N_R = 6$. . . . 97
A-12 Average regenerator number vs physical impairment threshold for lattice network with $N = 49$ and $N_R = 12$. . . . 97
A-13 Running time vs physical impairment threshold for NSFNET network with $N_R = 6$. 98
A-14 Running time vs physical impairment threshold for lattice network with $N = 49$ and $N_R = 12$. . . . 98
A-15 Acceptance Ratio vs total number of regenerators for NSFNET network with physical impairment threshold $= 1$. . . . 99
A-16 Acceptance Ratio vs total number of regenerators for lattice network with $N = 49$ and physical impairment threshold $= 1$. . . . 99
A-17 Average regenerator number vs total number of regenerators for NSFNET network with physical impairment threshold $= 1$. . . . 100
A-18 Average regenerator number vs total number of regenerators for lattice network with $N = 49$ and physical impairment threshold $= 1$. . . . 100
A-19 Running time vs total number of regenerators for NSFNET network with physical impairment threshold $= 1$. . . . 101
A-20 Running time vs total number of regenerators for lattice network with $N = 49$ and physical impairment threshold $= 1$. . . . 101

Acronyms

ASE Amplifier Spontaneous Emission

BER Bit Error Rate

BF Best Fit

CFPR Conversion Free Primary Routing

DLE Dynamic Lightpath Establishment

D_EISRA Dedicated Exact Impairment-aware Survivable Routing Algorithm

D_H1ISRA Dedicated Heuristic1 Impairment-aware Survivable Routing Algorithm

D_H2ISRA Dedicated Heuristic12 Impairment-aware Survivable Routing Algorithm

EDFA Erbium Doped Fiber Amplifier

DRPB Dedicated Regenerators for Primary and Backup

FF First Fit

HA Heuristic Algorithm

ISI Inter Symbol Interference

ISR Impairment-aware Survivable Routing

ISRPP Impairment-aware Regenerator Placement Problem

ISRP Impairment-aware Survivable Routing Problem

JPS Joint Path Selection

LLR Least Loaded Routing

M_SAMCRA Modified SAMCRA

MPI Multiple Physical Impairment

MPI_DRPB Multiple Physical Impairment DRPB

MPI_SRBB Multiple Physical Impairment SRBB

MPI_SRPB Multiple Physical Impairment SRPB

OEO Optical-Electronic-Optical

OSNR Optical Signal to Noise Ratio

OXC Optical Cross-Connect

PMD Polarization Mode Dispersion

POTS Plain Old Telephone Service

RF Random Fit

RWA Routing and Wavelength Assignment

SAMCRA Self-Adaptive Multi-Constrained Routing Algorithm

SLA Service Level Agreement

SLD Scheduled Lightpath Demand

SLE Static Lightpath Establishment

SPS Separate Path Selection

SRBB Shared Regenerators between Backups

SRPB Shared Regenerators between Primary and Backup

WDM Wavelength Division Multiplexed

Chapter 1

Introduction

1-1 Basic Concepts in WDM Optical networks

WDM optical networks are being widely used in long haul and metro/local networks. In transparent all-optical networks, the signal is transmitted from the source node to the destination node in the optical domain without any conversion to the electrical domain. But, if the signal is not regenerated at intermediate nodes along its route, noise and signal distortions are accumulated along the physical path. Each component at an intermediate node may introduce insertion loss. The optical fiber amplifiers, e.g., Erbium Doped Fiber Amplifiers (EDFAs), may compensate for some loss but also introduce noise at the same time. In addition, the Optical Cross-Connects (OXCs) and (de)multiplexers introduce crosstalk among different wavelength channels.

In order to go beyond the transparent reach of an optical signal, signal regeneration is required at intermediate nodes along the route to re-amplify, re-shape and re-time the optical signal, which are collectively known as 3R regeneration.

Signal regeneration is achieved through optical to electrical and then back to optical (O-E-O) conversions. In opaque networks, signal regeneration is employed at each node; whereas, in translucent networks, a compromise between opaque and all-optical networks, sparse regeneration is employed in which only some nodes have regeneration capability. Since regenerators are costly, the latter is preferred for practical implementations.

Core networks based on WDM technology constitute a promising and viable solution to support emerging applications requiring high availability and reliability guarantees. Due to the enormous bandwidth offered by these networks and the increasing number of mission critical applications, survivability is becoming an essential network design aspect.

Survivability is provided by the establishment of spare lightpaths for each connection request to protect the working lightpaths.

1-1-1 Routing and Wavelength Assignment (RWA)

In traditional communication networks, routing generally involves the identification of a path for each connection request between a source and destination node in the network. In optical networks, the wavelength of the path should also be determined. The resulting problem is called the Routing and Wavelength Assignment (RWA) problem. If wavelength conversion is allowed in the network, a lightpath can exit an intermediate node on a different wavelength. If no wavelength conversion is allowed then the wavelength continuity constraint is imposed to the generic RWA problem. This constraint implies that a lightpath should occupy only a specific single wavelength, throughout its path from the source to the destination node.

1-1-1-1 Types of Network

Based on their wavelength conversion capability WDM networks are divided into three types of networks:

- **A network with wavelength continuity constraint:** In a wavelength continuous network, a lightpath has to stay on the same wavelength on all the links it traverses. The wavelength assignment scheme will affect the network performance in such a network. A variety of wavelength assignment schemes have been studied in [39]. The wavelength selection can be performed in various ways, such as first fit (FF), best fit (BF) and random fit (RF). In first fit, each wavelength has a number associated with it, and the searching starts from the lowest/highest-numbered wavelength and stops as soon as an available wavelength has been found or all the wavelengths have been searched [27, 28]. The best fit approach tries to look through all of the candidate wavelengths so as to find the most appropriate [28, 29]. In the random fit approach, a wavelength is randomly chosen among the available wavelengths [30].

- **A network with full wavelength conversion capability:** A WDM network with full wavelength conversion capability can be considered to be equivalent to a conventional network, because wavelength assignment in such a network is a non-critical issue. In this network, we assume that each node in the network has a wavelength converter, which can convert any wavelength to any other wavelength. Wavelength assignment will not be a concern in such a network since a request will be satisfied as long as there is a free wavelength on the link.

- **A network with sparse wavelength conversion capability:** In a WDM network, if only a fraction of the nodes have a wavelength converter, we call the network has a sparse wavelength conversion capability. This has received much attention recently, because it can significantly reduce the number of wavelength converters. It also offers a flexible solution for the network carriers to upgrade their network gradually to support wavelength conversion.

1-1-1-2 Types of Traffic

The type of traffic also affects the choice of survivability techniques. Typically, the traffic demands can be static, dynamic or scheduled [31].

- **Static Lightpath Establishment (SLE):** traffic demands are known a priori independent of the current traffic condition in the network and traffic variations occur over long time scales. This is generally associated with design problems when network resources are allocated for the given input of traffic demands [21].
- **Dynamic Lightpath Establishment (DLE):** the connection requests arrive and depart in a random fashion or follow a certain pattern. This is generally what happens in general networks where the traffic is not known a priori. Once the network is provisioned, the critical issue is how to operate the network in such a way that the network performance is optimized under dynamic traffic [20, 21, 22, 23].
- **Scheduled lightpath demands (SLD):** the number of demands between a node pair and their set-up and tear-down times are known beforehand. In real optical transport networks, the traffic load is fairly predictable because of its periodic nature and this generally fits to the description of scheduled lightpaths [28].

1-1-2 Physical Layer Impairments

Signal impairments accumulate along a transparent optical path; therefore limiting the system reach and the overall network performance. The noise and signal distortions are known as physical impairments and degrade the quality of the received signal. In transparent optical networks for long distances and high bit rates the signal degradation may lead to an unacceptable bit error rate (BER). In such cases, it is necessary to regenerate the signal at intermediate nodes. In order to overcome physical impairments, 3R regeneration (re-amplification, re-shaping, and re-timing) is used.

In principle, optical 3R regeneration can be accomplished completely in the optical domain, but only electrical 3R regenerators are currently economically viable. Therefore, signal regeneration involves OEO conversion which disrupts the transparency of the signal. In transparent all-optical networks, the signal is transmitted in the optical domain from the source node to the destination node without undergoing any OEO conversions. In opaque networks, the optical signal carrying traffic undergoes an OEO conversion at every node; whereas, sparse regeneration is employed in translucent networks where only some nodes have regeneration capability. Since OEO process increases cost of signal transmission, sparse regeneration is preferred for practical implementations. Our study is thus focused on the regenerator placement problem in survivable translucent networks, where given a set of requests, the total number of regenerators required to accept these requests are minimized (resource usage minimization) and survivable routing in translucent networks where regenerators are already placed.

Physical impairments can be classified into two: linear and non-linear impairments.

- **Linear Impairments:** are independent of signal power and affect wavelengths individually, thus they can be handled as constraints associated to links or paths. Polarization mode dispersion (PMD), amplifier spontaneous emission (ASE) noise and chromatic dispersion are some of the examples of linear impairments.

  1. PMD is caused by the time delay between two orthogonal polarizations of light traveling at different speeds through an optical fiber. As a signal passes through EDFAs the optical signal to noise ratio (OSNR) is always degraded. PMD management requires that the time-average differential group delay between the two orthogonal states of polarization be less than a fraction a of the bit duration $T = 1/B$, where B is the bit rate. Typically, a is 10%. If the transparent segments consist of i spans, where the k^{th} span has fiber length $L(k)$ and fiber PMD parameter $D_{PMD}(k)$ measured in $ps/\sqrt{km}$, the constraint on differential delay can be expressed as [48]:

  $$B\sqrt{\sum_{k=1,\dots,i} D_{PMD}(k)^2 * L(k)} < a \tag{1-1}$$

  2. ASE is the dominant noise source in optical networks. The more amplifiers (EDFAs) an optical signal traverses, the higher ASE noise power it suffers from. Different models have been proposed to model a chain of optical amplifiers [47]. One model is the constant signal power model, which adjusts the gain of the amplifiers so that a constant signal power is maintained at the output of every amplification span. In this scenario, the noise after the i^{th} amplification span is:

  $$n_{out,i} = n_0 + \sum_{k=1,\dots,i} 2n_{sp} * h * v_s(G_k - 1) * B_0 \tag{1-2}$$

  where n_0 is the initial noise at the input, $n_{out,i}$ is the noise after the i^{th} amplification span, n_{sp} is the excess noise factor, h is the Planck's constant, v_s is the carrier frequency, B_0 is the optical bandwidth, and G_k is the gain of the k^{th} amplifier. Thus, the ASE noise on a link can be modeled as the sum of the ASE noise across spans, and similarly the end-to-end ASE noise.

3. Chromatic dispersion brings about pulse broadening, thereby affecting the receiver performance by reducing the pulse energy within the bit slot and spreading the pulse energy beyond the allocated bit slot leading to inter-symbol interference (ISI).

- **Non-linear Impairments:** are significantly more complex than their linear counterparts and in addition to generating dispersion on each channel they also create crosstalk between channels. In particular, non-linearities strongly depend on the current allocation of wavelengths on a given fiber (and path), and therefore on the current status of allocated lightpaths in addition to the physical topology. This intuitively affects the routing and wavelength assignment (RWA) problem solution of new lightpath requests, i.e., the selection of a suitable path and wavelength may fail to meet the minimum transmission requirement. It may also affect already established lightpaths whose transmission properties are negatively affected by the new establishing lightpath.

Different types of link cost functions have been suggested in different papers to represent physical impairments. In this thesis, additive linear physical impairments which are independent of the impairment cost function are considered.

1-1-3 WDM Optical Network Survivability

Recent advances in optical switching, and in particular, wavelength division multiplexing (WDM) have enabled next generation networks to be able to operate at several Tera bits per second [18, 19]. Wavelength routed optical networks consist of optical switching nodes interconnected by one or more fiber links. In such networks, failures (of links or nodes) may result in huge data losses due to the enormous bandwidth per fiber. Optical network survivability is defined as the ability of the optical WDM network to gracefully respond to such failures [37].

Service providers that deploy networks must adhere to service level agreements (SLAs) that bind them to meet the customer's requirements such as availability, speed and quality of service. For some applications that require uninterrupted services such as financial and medical data transfer, one of the most critical SLA involves the availability of the network. The impact of the network outage can be normally measured in terms of customer minutes defined as the outage in minutes multiplied by the number of affected customers. The standard of network availability set for plain old telephone services (POTS) is 99.999% (or 5-minute downtime per year). Some applications may further need a more stringent requirement (e.g., 99.9999% network availability). In both cases, the downtime of the network needs to be minimized by providing a fast restoration mechanism. This involves the fast recovery of traffic from the failure of the various elements of the network, including transmission medium (link) and equipment (node) failures.

Due to the frequent occurrence of fiber cuts and the tremendous amount of data transported, survivability, which is the ability to reconfigure and reestablish communication upon failure, is indispensable in WDM networks. Survivability is a key concern in modern network design, and has gained an increasing attention from both network carriers and researchers. WDM networks are usually employed as multi-layered networks, hence survivability can be provided either at the optical layer or by higher layers. However, the recovery time at higher layers is in the order of seconds (upon failure of a single fiber a number of logical failures may be detected at higher layers unnecessarily complicating and delaying the restoration mechanism), while at the optical layer it is only in the order of milliseconds. In addition, survivability at the optical layer is more efficient because of resource sharing and may provide survivability to higher layers which do not have inherent survivability capability [18, 25, 26].

When a component fails, all the lightpaths that are currently using this component will also fail. If the network is survivable, another lightpath which does not use the failed component will take over from the failed component. The lightpath that carries traffic during normal operations is known as the primary lightpath, whereas the lightpath that is used to reroute traffic when a primary lightpath fails is called the backup lightpath.

Survivability in optical networks can be realized by protection (pro-active) or by restoration (reactive) mechanisms. In protection based schemes, each incoming connection request is provided with a primary path and a link-disjoint backup path at setup time. In restoration based schemes, an alternate path is determined only after the failure occurs. In this thesis a proactive recovery mechanism is used. A typical approach to the design of survivable networks is through a protection scheme that pre-determines and reserves backup bandwidth considering single/multiple link/node failure scenarios. One of the key challenges in survivable optical networks is to devise strategies to determine primary and backup paths [37] such that the network throughput is maximized and resource consumption is minimized.

A failure scenario includes:

- **Link failure:** This type of faults often results from external causes, cable cuts are very frequent especially in terrestrial networks since fiber cables often share other utility transport conduits such as gas or water pipes and electrical cables. The link failure scenario is the most widely studied scenario. This is mainly due to two factors: it is more prevalent compared to other failures due to the high frequency of fiber cuts and the techniques used to protect against link failures can be extended for other failures, such as node failures. This also includes failure of link components such as line amplifiers and regenerators. Fiber cuts are a more likely cause for this type of failure. A single link failure model is usually assumed because it is easier to plan for the failure of at most one piece of equipment at a time. In the past, failures were manually solved by temporarily re-routing the broken connections and sending teams to repair the damaged equipment in site. Today, optical networks that still require manual rerouting can be considered as unprotected. The outage periods due to traffic recovery based on the human intervention are unacceptable, even if nowadays the apparatus of a digital telephone network can be remotely re-configured from an operative head quarters, a doubtless advantage compared to pre-digital telephone systems. At present no optical network operator is willing to accept unprotected facilities: survivability must be always guaranteed by adopting efficient techniques of automatic recovery from failures, that is to say re-routing broken connections automatically [19]. Link failures are further divided into single or multiple link failures.

  - **Single link failure:** In this kind of failure only a single link is affected among all the links of the network. This type of failure is the simplest one among the others in terms of implementation of recovery methods. In this thesis only single link failures will be considered.
  - **Multi-link failure:** In this kind of failure multiple links are affected at the same time during a failure. Compared to single link failure this type of failure's recovery methods are difficult to implement.

- **Node failure:** This type of failures occur mainly due to internal causes such as hardware degradation or management software inefficiency. They can result also from operator error or power outages as well as other disasters that lead to component failures in a node. Operator errors are a more likely cause for this type of failure. However, forecasting statistically characterizing external causes is so difficult that they are not usually taken into account in network design. Although less common, node failures can have devastating consequences since they can interrupt all the connections that traverse the failed node and are traditionally more difficult to handle compared to link failures. Node failures are further divided into single or multiple node failures.
  - **Single node failure:** In this kind of failure only a single node in the network is malfunctioning. In case of a single node failure in a chain topology network, the chain will be divided into two separate chains. In a hub architecture, one of these chains will be completely disconnected. Ring topologies can recover from single node failure.
  - **Multi-node failure:** In this kind of failure multiple nodes will be affected at the same time.

Generally, there are three ways of recovering from a failure of the primary lightpath: link based, sub-path (segment) based and path based.

- **Link based protection/restoration:** A link based protection/restoration scheme provides protection or restoration for each link and a local detouring of the failed link is employed during a link failure. Unlike path protection where the end nodes of the connection need to be signaled to handle the failure, the backup path signaling is handled at the end nodes of the link. This will lead to a lower recovery time for link protection when compared to that of path protection. However, link protection is less flexible because the backup paths are usually long and fewer in number, and in wavelength-selective networks, the backup route must use the same wavelength as the primary route, since its working segment is retained [40].

- **Segment based protection/restoration:** The sub-path based protection/restoration scheme is a compromise between path based and link based protection/restoration schemes. In sub-path based protection, backup routes are precomputed for segments of the primary route. In sub-path based restoration, when a link fails, the upstream node of the failed link detects the failure and discovers a backup route from itself to the corresponding destination node for each disrupted connection. Segment protection can be used as an efficient way to avoid traps in a survivable WDM mesh network [19]. Compared to path protection, the sub-path protection can achieve smaller recovery time, since the signaling does not need to traverse the entire path back to the source to initiate the protection procedure. On the other hand, sub-path protection sacrifices resource utilization.

- **Path based protection/restoration:** In path based protection, link-disjoint or node-disjoint backup lightpaths are precomputed and take over when a primary lightpath fails. In path based restoration, the source and the destination nodes of each connection request (lightpath) affected by a failure run a distributed RWA algorithm to dynamically determine the backup path and wavelength(s) on an end-to-end basis. If the algorithm finds a free backup light path, the traffic is then routed on that path on appropriate wavelength(s) after signaling its cross-connects. If not, the connection is blocked [41, 43, 44]. Path protection attempts to restore a connection on an end-to-end basis by providing a backup path in case the primary path fails. The backup path assignment may be either independent or dependent on the link failure in the network. For example, a backup path that is link-disjoint with the primary path allows recovery from single link failures without the precise knowledge of failure location.

In terms of resource usage, protection schemes can also be divided as dedicated and shared.

- **Dedicated protection/restoration:** In this scheme, wavelength channels are not shared with any other backup or primary paths and are used only for the chosen connection request (1+1, 1:1). Switching from the working path to the corresponding restoration path is performed by exchanging signaling messages between the source destination pair. Dedicated protection is the easiest way to allocate capacity for backup paths, i.e., allocating the same amount as for the working paths. In 1+1 protection, traffic is transmitted simultaneously on both working and protection paths, and in case of failure, the destination is required to switch between working and protection paths, i.e., non-signaled switchover. Although such recovery is very fast, it is very inefficient due to the inherent resource redundancy. Unlike 1+1 protection, in 1:1 path protection both working and protection paths are provisioned simultaneously, but data is only rerouted to the working path. To utilize the resource efficiently the protection path in 1:1 can be used to transmit low priority pre-emptable traffic during non-failure conditions.

- **Shared protection/restoration:** In the shared scheme ($1 : N$), backup lightpaths may share wavelength channels on some links as long as their primary lightpaths do not share the same resource (the primary paths use mutually disjoint sets of links). The shared scheme provides a better resource utilization, however it is more complicated and requires more information, such as the share ability of each link, etc. Depending on the type of lightpaths (primary or backup lightpaths) that share the backup resources, we can have two schemes:

  1. Backup or shared multiplexing: Here the backup resources are shared between one or more backup paths that do not need them simultaneously.
  2. Primary backup multiplexing: In this case, the backup resources can be shared between a primary path and one or more backup paths that do not need them simultaneously [20, 45]. While this scheme reduces the connection blocking probability, it could result in a potential reduction in restoration guarantee. Moreover, this technique is suitable for a dynamic traffic scenario where lightpaths are setup and torn down frequently.

The problem of finding link-disjoint primary and backup paths is typically solved using two different approaches: separate path selection (SPS) or joint path selection (JPS).

- **Separate Path Selection (SPS):** This is the typical approach where the algorithm first selects the path with the minimum cost as the primary path and then selects a link-disjoint path with least cost as the backup.

- **Joint Path Selection (JPS):** In this approach, the algorithm tries to optimize the combined cost of the primary and the backup paths. SPS approaches that take into account the current network state (e.g. LLR and Conversion Free Primary Routing (CFPR)) were seen to perform better than the basic hop count scheme. However, a technique that tries to optimize the combined cost of the primary and backup paths has been shown to perform even better [38, 39].

In this thesis, we will consider path based protection, which many previous studies considered to be easier to implement. In path based protection, a protection path is used to prevent the services on the working path from disruption. The working path and protection path must be disjoint to avoid any single point of failure, e.g., a fiber cut. In shared-path protection, protection paths of different connections may share the same wavelength on a link as long as their working paths are not subject to a single point of failure.

1-2 Related Work

Recently there has been an increasing interest in impairment-aware routing problems. Most of the works study the problem of finding feasible paths that satisfy a given set of impairment constraints. A detailed survey of impairment-aware RWA algorithms is given by Azodolmolky *et al.* [38].

Several strategies have been considered to design a translucent optical network. In [2, 3], the problem of designing a survivable translucent optical network was formulated as an integer linear programming problem (ILP). In [4], the traditional resource sharing scheme, i.e., sharing of a wavelength, was extended to include sharing of regenerators. In [5], the authors presented an ILP formulation and a local optimization heuristic approach with an objective of minimizing the wavelength links and regenerators used. In [6] they studied the problem of selecting wavelength converter and regeneration sites in translucent network. The wavelength converter placement problem was formulated using an integer linear program and several heuristic algorithms for solving the sparse wavelength converter and regeneration placement problem were proposed.

In reference [8], Suurballe proposes an algorithm, referred to as Suurballe's algorithm, to find K disjoint paths with minimal total length using a path augmentation method. Bhandari's algorithm [7] was designed to find a pair of span-disjoint paths in an optical network.

In [9] DIMCRA, a heuristic algorithm for multi-constrained link-disjoint path pair, is proposed. Given a weighted digraph G and an m-dimensional constraint vector C, the main steps of DIMCRA are the following. It finds the shortest path P_1 obeying C in G. Then it reverse all links on P_1 and set their link weights to zero making graph G'. Then, it finds the shortest path P_2 constrained by $2C$ in graph G'. It then removes the interlacing links of P_1 and P_2 and check feasibility of the paths. If path P_i $(i = 1, 2)$ violates the constraints, update G' by removing the link set $P_i - (P_i \cap P_1)$ from it, and find P_2. DIMCRA has a link removing operation, which makes it unable to always find the existing feasible solution, let alone the optimal one.

In [10] the authors addressed the problem of translucent network design by proposing several regenerator placement algorithms based on different knowledge of future network traffic patterns. They also addressed the problem of wavelength routing under sparse regeneration by incorporating two regenerator allocation strategies with heuristic wavelength routing algorithms. In addition, they proposed network topology based regenerator placement and traffic prediction based regenerator placement algorithms. In the network topology based regenerator placement, regeneration demands are most likely to be generated at two categories of nodes. The first category consists of those nodes that are located at the center of a network. The second category consists of those nodes that have a higher nodal degree than other nodes. The traffic prediction based regenerator placement algorithms favor the nodes with heavier traffic loads and the nodes with more through lightpaths suffering signal quality degradation.

There are also works that study the optimal placement of regenerators in a network [14, 15, 16, 17, 18, 19, 20, 21]. Some of these works depend on the type of physical impairment and some of them are specific to certain networks. In [36, 11] the authors have studied the regenerator placement problem with the objective of finding the minimum number of nodes, where regenerators are to be placed so that there is a feasible path between any pairs of nodes in the given network. They have shown that this problem is NP-complete and have provided heuristic algorithms. Flammini *et al.* [12] have considered different variants of the regenerator placement problem under the assumption that all links have the same cost, thus the impairment threshold is basically determined by the hop count of the path.

The main goal in the aforementioned regenerator placement studies is to minimize the number of nodes where regenerators are placed. However, minimizing the number of regenerator nodes does not necessarily minimize the total number of regenerators needed. Therefore, in this thesis, we study how to sparsely place regenerators in the network to minimize the number of regenerators and survive network failures at the same time, and impairment-aware survivable routing where the main objective is to minimize the total number of regenerators used in the network.

It is of interest to consider a variant of the impairment-aware routing problem, in which only simple paths are admitted as solutions; indeed, such restrictions may be due to scarcity of resources (link or node capacity) or management considerations. We will address the problem of survivable lightpath provisioning in a resource efficient manner using simple paths, i.e., minimizing number of regenerators required. Unlike most of the works discussed earlier, in addition to assigning dedicated regenerators for the primary and backup paths of a request, a resource sharing scheme that supports the following kinds of resource sharing scenarios will also be proposed and implemented:

- Regenerator sharing between working and backup lightpaths and
- Regenerator sharing between backup lightpaths

These scenarios are from the fact that not only can wavelength links be shared, but the regenerators can also be shared by the protection paths if their working paths are link-disjoint. In addition, regenerators can also be shared between a working path and its protection path. This is because the working and protection paths do not need regeneration at the same time in the case of 1:1 protection.

Because a signal is regenerated only if necessary, we need much fewer regeneration resources. This thesis addresses the survivable network design and survivable routing problems in translucent optical networks. We refer to the translucent network design problem as the survivable regenerator placement problem while the routing problem as impairment-aware survivable routing problem.

1-3 Objectives

In this thesis, both survivable regenerator placement and physical impairment-aware survivable routing problems are studied and solutions are provided.

Most of the regenerator placement related works mentioned before focus on in minimization of the number of nodes where regenerators are placed and the others related to physical impairment-aware routing do not integrate regenerator assignment in the path computation process of the routing algorithm. Therefore, this thesis mainly differs from the previous works in that it minimizes the number of regenerators rather than the number of nodes where regenerators are placed for the survivable regenerator placement problem and it incorporates the regenerator assignment in the path computation process of the physical impairment-aware survivable routing problem.

Due to the physical impairments which deteriorate the quality of the signal 3R regeneration of signals in WDM networks is necessary which involves OEO conversions. In addition, since optical fibers carry a large amount of data and frequent the occurrence of failure makes survivability indispensable in these type of networks. Therefore, the effect of physical impairments and link failures in WDM networks lead to the objectives of this thesis work.

The main objectives of the thesis are:

- To develop survivable regenerator placement algorithms and investigate their performances.
- To develop impairment-aware survivable routing algorithms in translucent optical networks and investigate their performances.

1-4 Organization of the Thesis

The remaining part of the thesis is organized as follows. In chapter 2, survivable regenerator placement problems are defined and we present both exact and heuristic algorithms that solve the problems along with simulation results and performance analysis of the algorithms.

In chapter 3, after defining impairment-aware survivable routing problems, algorithms that solve them with their simulation results and performance analysis are given. Finally, in chapter 4, we give a conclusion on the work done and make some recommendations for future work.

Chapter 2

Survivable Regenerator Placement Problem

The physical impairments of an optical medium limit the number of network elements that can be traversed by an optical signal without making use of regenerators. We define a regenerator segment of a lightpath to be a transparent segment (i.e., one or more links) between two regenerator nodes (including source and destination nodes) of the lightpath. A lightpath can be made up of multiple regenerator segments. After a signal is regenerated, its original physical features are restored. Thus, from a physical impairment point of view, the effect of physical impairments along the path followed to reach the regenerator node is completely removed.

In the design of translucent networks, we deal with the regenerator placement problem, which distinguishes a translucent network design problem from a conventional transparent network design problem. To focus on this problem, we assume that the network topology and optical layer components (e.g., fibers, amplifiers, OXCs, and (de)multiplexers) have already been deployed.

Assume that both the optical topology and deployed optical layer components are given. Then, a cost function to be minimized is the number of regenerators, which are required in the given optical topology to provide end-to-end connectivity. Minimizing number of regenerators in a network reduces initial cost, operation cost and maintenance cost of regenerators. In addition, power consumption, heat dissipation and space requirements will also be reduced since the electronic processing is reduced.

In this chapter, the survivable regenerator placement problem with an objective of minimizing the number of regenerators required will be solved. In this problem, it is assumed that there are enough wavelengths so that the wavelength continuity constraint is relaxed.

The survivable regenerator placement problem is further divided into three sub-problems: dedicated survivable regenerator placement problem, shared survivable regenerator placement problem and backup shared survivable regenerator placement problem. The problem definitions, algorithms used to solve the problems and simulation results that show the performance of the algorithms are given in upcoming sections.

2-1 Dedicated Survivable Regenerator Placement Problem

In this sub-problem, dedicated regenerators are placed for the primary and backup lightpaths of a request. Dedicated assignment of regenerators is used in a scenario in which the network is protected by the 1+1 protection mechanism. Since we are considering dedicated protection, the different requests do not share regenerators. Thus, each request can be considered individually. i.e., the problem is reduced to a single request problem.

2-1-1 Problem Definition

1. **Given:** A network topology which is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of N nodes and $\mathcal{L}$ is the set of L links between the nodes; associated with each link $(u, v) \in \mathcal{L}$ there is a wavelength and and a set of non-negative physical impairment metrics $r_i(u, v)$ for $i = 1, \ldots, m$. Given is a request represented by the tuple (s, d, Δ), where $s, d \in \mathcal{N}$ are the source and destination nodes of the request and Δ is the set of threshold values of the physical impairments.

2. **Objective:** Find a pair of link-disjoint simple paths from the source node to the destination node and place dedicated regenerators for the primary and backup paths, minimizing the total number of regenerators used by the two paths.

To solve the dedicated survivable regenerator placement (DSRP) problem, we provide an exact integer linear program (ILP) formulation using network flow equations and a heuristic algorithm that uses concepts of Suurballe's link-disjoint path computation algorithm.

2-1-2 Exact Algorithm

We formulate the dedicated survivable regenerator placement problem for a given request using an ILP. An ILP may take an exponential amount of time to obtain an optimal solution, which makes it inconvenient for large networks. The formulation of the exact algorithm, Exact, is given as follows:

Sets:
$\mathcal{L}^+(u)$ Set of outgoing links of node u
$\mathcal{L}^-(u)$ Set of incoming links of node u

Variables:

$x_{l,u}$ - is 1 if the primary lightpath uses link l and node u is the last regenerator node or the source node before encountering link l; 0 otherwise.

$y_{l,u}$ - is 1 if the backup lightpath uses link l and node u is the last regenerator node or the source node before encountering link l; 0 otherwise.

$\gamma_{u,v}$ - is 1 if the primary lightpath uses a regenerator at node u (or u is a source node) followed by a regenerator at node v.

$\tau_{u,v}$ - is 1 if the backup lightpath uses a regenerator at node u (or u is a source node) followed by a regenerator at node v.

Objective:
Minimize the total number of regenerators needed by the primary and backup lightpaths. Minimize:

$$\sum_{u \in \mathcal{N}} \sum_{v \in \mathcal{N}} (\gamma_{u,v} + \tau_{u,v}) \tag{2-1}$$

Constraints:
Flow conservation constraints:
At the source node there are exactly two flows leaving the source node, one for the primary and another for the backup lightpaths.

$$\sum_{l \in \mathcal{L}^+(s)} (x_{l,s} + y_{l,s}) = 2 \tag{2-2}$$

At the destination node there are exactly two flows entering the destination node, one for the primary and another for the backup lightpaths.

$$\sum_{l \in \mathcal{L}^-(d)} \sum_{u \in \mathcal{N} \backslash d} (x_{l,u} + y_{l,u}) = 2 \tag{2-3}$$

At intermediate nodes, a flow that enters a node has to leave it after being regenerated or not.

$$\sum_{l \in \mathcal{L}^-(v)} x_{l,u} - \sum_{l \in \mathcal{L}^+(v)} x_{l,u} = \gamma_{u,v} \quad and \quad \sum_{l \in \mathcal{L}^-(v)} y_{l,u} - \sum_{l \in \mathcal{L}^+(v)} y_{l,u} = \tau_{u,v}; \quad \forall v \in \mathcal{N} \backslash \{s,d\};\ \forall u \in \mathcal{N} \backslash v \tag{2-4}$$

If a lightpath is regenerated at node v, the last regenerator node in the new segment should be node v, and not any other node.

$$\sum_{l \in \mathcal{L}^+(v)} x_{l,v} - \sum_{u \in \mathcal{N} \backslash v} \gamma_{u,v} = 0 \quad and \quad \sum_{l \in \mathcal{L}^+(v)} y_{l,v} - \sum_{u \in \mathcal{N} \backslash v} \tau_{u,v} = 0; \qquad \forall v \in \mathcal{N} \backslash \{s, d\} \tag{2-5}$$

Disjointedness constraints:
The primary and backup lightpaths should be link-disjoint.

$$\sum_{u \in \mathcal{N}} (x_{l,u} + y_{l,u}) \leq 1; \quad \forall l \in \mathcal{L} \tag{2-6}$$

For undirected networks, we first replace each link with two directed links in either direction. Let for each $l = (u, v) \in \mathcal{L}$, its corresponding oppositely directed link be $l' = (v, u) \in \mathcal{L}$. Then equation 2-6 becomes:

$$\sum_{u \in \mathcal{N}} (x_{l,u} + y_{l,u} + x_{l',u} + y_{l',u}) \leq 1; \quad \forall l \in \mathcal{L} \tag{2-7}$$

Simple path constraints:
The lightpaths should not contain loops.
At the source node, there should not be a flow associated with any of its incoming links.

$$\sum_{l \in \mathcal{L}^-(s)} \sum_{u \in \mathcal{N} \backslash s} (x_{l,u} + y_{l,u}) = 0 \tag{2-8}$$

At the destination node, there should not be a flow associated with any of its outgoing links.

$$\sum_{l \in \mathcal{L}^+(d)} \sum_{u \in \mathcal{N} \backslash d} (x_{l,u} + y_{l,u}) = 0 \tag{2-9}$$

Any flow that exits the source node, other than the one originating at the source node, should explicitly be set to 0.

$$\sum_{l \in \mathcal{L}^+(s)} \sum_{u \in \mathcal{N} \backslash s} (x_{l,u} + y_{l,u}) = 0 \tag{2-10}$$

For any intermediate node there can at most be one flow of the primary or backup lightpath entering the node.

$$\sum_{l \in \mathcal{L}^-(v)} \sum_{u \in \mathcal{N}} x_{l,u} \leq 1 \quad and \quad \sum_{l \in \mathcal{L}^-(v)} \sum_{u \in \mathcal{N}} y_{l,u} \leq 1; \qquad \forall v \in \mathcal{N} \backslash s \tag{2-11}$$

Impairment constraints:
The physical impairment of any transparent segment should be less than the threshold.

$$\sum_{l \in \mathcal{L}} r(l) * x_{l,u} \leq \Delta \quad and \quad \sum_{l \in \mathcal{L}} r(l) * y_{l,u} \leq \Delta; \quad \forall u \in \mathcal{N} \tag{2-12}$$

2-1-3 Single Physical Impairment Heuristic Algorithm

While ILPs are useful in providing insights into the nature of the problem and the solutions obtained are optimal, they are applicable only for relatively small sized networks; larger network sizes demand excessive computation, as the complexity of the ILP algorithms increases exponentially with the network size. As a result, network design based on heuristic algorithms is usually considered as a more effective means to solve large scale network problems. A heuristic algorithm, DRPB (dedicated regenerator for primary and backup), that uses concepts of Suurballe's algorithm is given below.

Algorithm 1 DRPB(G, s, d, Δ)

1: Find the shortest path $\{P^*_{u\to v}\}$ between all nodes $u, v \in \mathcal{N}$, for which $r(P^*_{u\to v}) \leq \Delta$.
2: Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' = \{(u, v) \mid r(P^*_{u\to v}) \leq \Delta\}$ and assign a cost of 1 to each link.
3: Find the shortest path $P'_{s\to d}$ from s to d in G'.
4: Substitute the links of $P'_{s\to d}$ with the corresponding sub-paths $P^*_{u\to v}$ in G to obtain $P_{s\to d}$.
5: Remove all the loops of $P_{s\to d}$ in graph G to obtain path $P^1_{s\to d}$.
6: Redirect all links in $P^1_{s\to d}$ from d to s to obtain $G''(\mathcal{N}, \mathcal{L}'')$.
7: On graph G'' repeat steps $1-5$ to obtain path $P^2_{s\to d}$.
8: Remove all links that are both in $P^1_{s\to d}$ and $P^2_{s\to d}$ to obtain two link-disjoint paths.
9: Place dedicated regenerators for each path.

In step 1 of DRPB, the shortest distance between all nodes is computed using Dijkstra's algorithm in graph G and if the distance computed in step 1 is less than or equal to the physical impairment threshold, a link connecting the nodes will be added in step 2 while constructing graph G'. i.e., all nodes that are within a distance of less than or equal to the physical impairment threshold will be directly connected in graph G'. Hence, the links in graph G' represent sub-paths in graph G. The approach of creating a new graph by connecting nodes that are within a distance of less than or equal to the physical impairment threshold is chosen to minimize the number of regenerators. The nodes in graph G' are the possible places where regenerators are placed. Therefore, finding the shortest path in graph G' is finding a path that has the minimum number of regenerators.

Once the shortest path is obtained in step 3, the path is transformed to its equivalent path $P_{s\to d}$ in graph G in step 4. However, since this path is made of a concatenation of path segments, it may not be a simple path. Hence its loops are removed in step 5 and the links along the loopless path $P^1_{s\to d}$ are redirected from d to s to obtain graph G'' in step 6. Note that, for undirected graphs, the directed links in G'' may result in cases where $P^*_{u\to v} \neq P^*_{v\to u}$, in which case the graph obtained from G'' may contain two directed links between nodes u and v, one in either direction. In step 7, the same procedures are repeated in graph G'' to find a second loopless path, $P^2_{s\to d}$. Once the second path $P^2_{s\to d}$ is computed, the interlacing links between $P^1_{s\to d}$ and $P^2_{s\to d}$ are removed to get link-disjoint primary and backup paths. Finally, the regenerators are placed accordingly on these paths.

Complexity of DRPB

The major operations that determine the complexity of algorithm DRPB are: constructing graphs G' and G'', finding the shortest paths between s and d in these graphs and removing the loops of the paths in G. The construction of the graphs involves finding the shortest paths between each pair of nodes. This can be implemented with complexity of $O(N^2 \log N + NL)$ as shown in [13]. Since the computation complexity of Dijkstra's algorithm is $O(N^2)$, finding the shortest paths in G' and G'' will have a complexity of $O(N^2)$. The paths obtained in G' and G'' have $O(N)$ hop count in the worst case and at each node there can be a loop with $O(N)$ hop count in graph G, thus the total complexity of removing loops is $O(N^2)$. Therefore, the total complexity of algorithm DRPB is $O(N^2 \log N + NL)$.

2-1-4 Simulation Results

This sub-section discusses the simulation results of the dedicated algorithms discussed earlier in this chapter. In our simulations, the main objective is to find link-disjoint primary and backup paths for a single connection request and place minimum regenerators for both paths. In addition, the goal of the simulation is to get a better insight into the performance of the algorithms and make a comparison between the algorithms while varying the physical impairment threshold.

In order to achieve the objectives, different scenarios are simulated. The simulations of the algorithms discussed in this chapter as well as in the next chapter are performed on lattice networks of $N = 5X5$ and $N = 7X7$ nodes, random networks of $N = 25$ and $N = 49$ nodes with link probability of $\rho = 0.2$ which represents the probability of existence of a link between any pair of nodes in the network, an ARPANET network with $N = 28$ and $L = 45$ and an NSFNET network with $N = 14$ and $L = 21$. These networks are shown in Figure 2-1 and Figure 2-2. In our simulations, we used the terms Δ and $Rmax$ interchangeably to represent the physical impairment threshold.

In our simulations in this thesis, the link costs, i.e., the physical impairments are randomly generated in the range of (0,1] by a uniformly distributed random function and the source and destination are also randomly selected for each request.

The algorithms given earlier and the ones that follow in the upcoming sections are implemented using a C programming language. We utilized the lpsolve software package to solve the instances of the ILPs generated for a representative network topology.

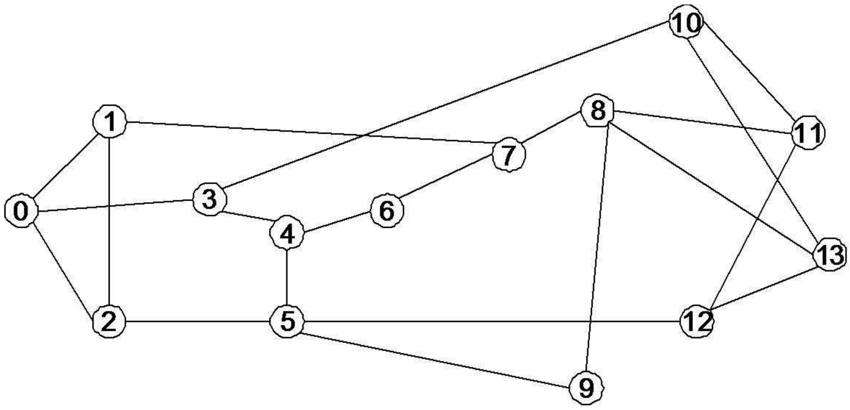

Figure 2-1: The NSFNET network.

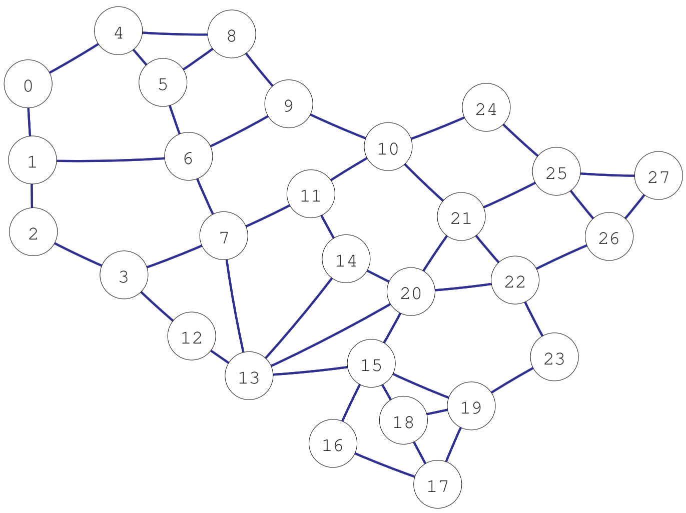

Figure 2-2: The ARPANET network.

The simulation results are shown on the graphs of Figure 2-3 to Figure 2-5. The performance of the algorithms is compared based on the average regenerator number needed by both the primary and backup paths of a request by varying the physical impairment threshold. As can be seen from the results, as the threshold value of the physical impairment increases the number of regenerators required for the request is reduced. This result is obtained due to the fact that, for a large physical impairment threshold a signal will travel longer distance before violating the physical impairment threshold, i.e., the signal travels longer without needing regeneration, which leads to reduction of the number of regenerators required. The results also show that, the solution obtained from the heuristic algorithm is the same as the one obtained from the exact algorithm for lattice and random networks; whereas, it is slightly different for the ARPANET network.

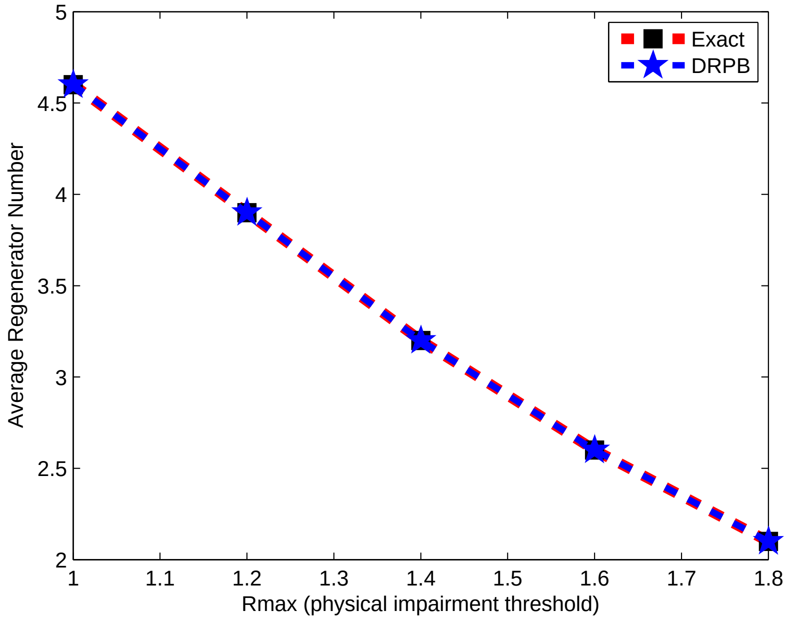

Figure 2-3: Average regenerator number vs physical impairment threshold for lattice network with $N = 25$ (Exact vs DRPB).

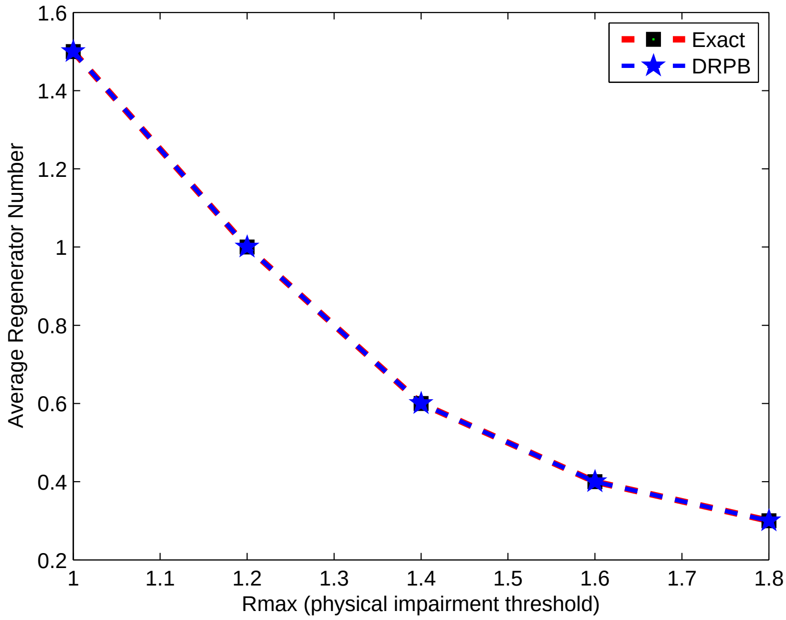

Figure 2-4: Average regenerator number vs physical impairment threshold for random network with $N = 25$ (Exact vs DRPB).

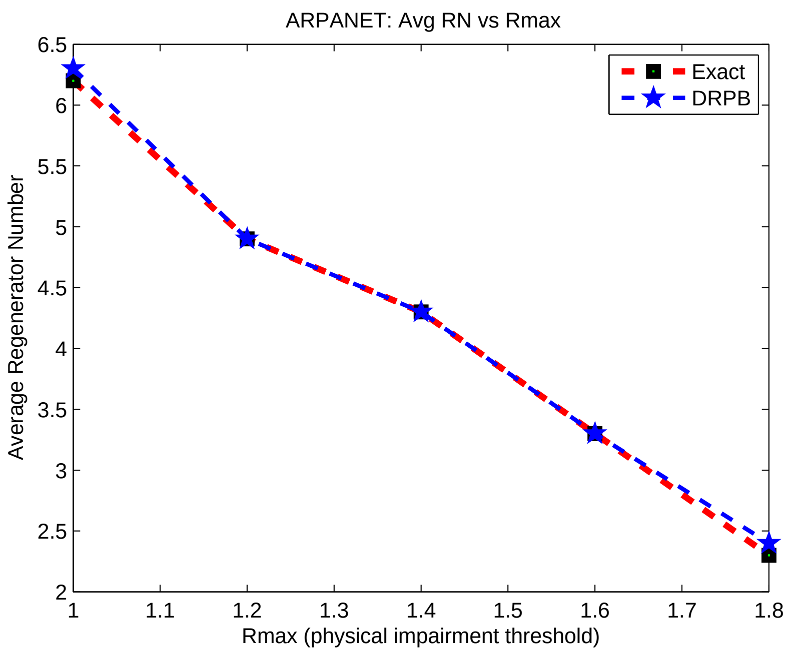

Figure 2-5: Average regenerator number vs physical impairment threshold for the ARPANET network (Exact vs DRPB).

2-2 Shared Survivable Regenerator Placement Problem

In this sub-problem, regenerators are shared between the primary and backup lightpaths of the same request. This type of regenerator assignment is used in a network protected by the 1:1 protection mechanism, since the primary and the backup paths do not need regeneration at the same time. Since regenerators are only shared between primary and backup paths of the same request, each request is considered independently. i.e., the problem becomes shared survivable regenerator placement problem for a single request.

2-2-1 Problem Definition

1. **Given:** The physical optical network is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of N nodes and $\mathcal{L}$ is the set of L links between the nodes; associated with each link $(u, v) \in \mathcal{L}$ there is a wavelength and and a set of non-negative physical impairment metrics $r_i(u, v)$ for $i = 1, \dots, m$. A request is represented by the tuple (s, d, Δ), where $s, d \in \mathcal{N}$ are the source and destination nodes of the request respectively and Δ is the set of threshold values of the physical impairments.
2. **Objective:** Find a pair of link-disjoint simple paths from the source node to the destination node and place shared regenerators between the primary and backup paths, minimizing the total number of regenerators used by the two paths.

To solve the shared survivable regenerator placement problem, we provide an exact integer linear program (ILP) algorithm (Exact_S) and a heuristic algorithm (SRPB - shared regenerators between primary and backup). The algorithms are given below.

2-2-2 Exact Algorithm

The ILP formulation of the exact algorithm (Exact_S) is the same as the exact algorithm given earlier for the dedicated survivable regenerator placement problem case, but with a different objective function and one additional constraint. Instead of repeating all the formulations only the additional formulations are given below.

Variables:

α_u - is 1 if a regenerator (shared or not) is needed at node u; 0 otherwise.

Objective:
Minimize the total number of regenerators needed by the primary and backup lightpaths.
Minimize:

$$\sum_{u \in \mathcal{N}} \alpha_u \tag{2-13}$$

Constraint:
If a regenerator is needed at a node, it is needed by the primary lightpath or the backup lightpath or both the primary and backup lightpaths.

$$\sum_{u \in \mathcal{N}} (\gamma_{u,v} + \tau_{u,v}) \leq 2 * \alpha_u; \quad \forall v \in \mathcal{N} \tag{2-14}$$

2-2-3 Single Physical Impairment Heuristic Algorithm

As discussed earlier, the number of variables and the number of equations for ILPs grow rapidly with the size of a network. Therefore, the ILP formulations are practical only for small sized networks. For larger networks, we need to employ heuristic methods. Therefore, we provide a simple heuristic algorithm which efficiently increases regenerator sharing between primary and backup paths.

Algorithm 2 SRPB(G, s, d, Δ)

1: Find the shortest path $\{P^*_{u\to v}\}$ between all nodes $u, v \in \mathcal{N}$, for which $r(P^*_{u\to v}) \leq \Delta$.
2: Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' = \{(u, v) \mid r(P^*_{u\to v}) \leq \Delta\}$ and assign a cost of 1 to each link.
3: Find the shortest path $P'_{s\to d}$ from s to d in G'.
4: Substitute the links of $P'_{s\to d}$ with the corresponding sub-paths $P^*_{u\to v}$ in G to obtain $P_{s\to d}$.
5: Remove all the loops of $P_{s\to d}$ in graph G to obtain path $P^1_{s\to d}$.
6: Redirect all links in $P^1_{s\to d}$ from d to s to obtain $G''(\mathcal{N}, \mathcal{L}'')$.
7: Temporarily place regenerators, T_R, for path $P^1_{s\to d}$.
8: Find the shortest path $\{P^*_{u\to v}\}$ between all nodes $u, v \in T_R$ in graph $G''(\mathcal{N}, \mathcal{L}'')$.
9: Make a graph $G'''(\mathcal{N}, \mathcal{L}''')$, where $\mathcal{L}''' = \{(u, v) \mid r(P^*_{u\to v}) \leq \Delta\}$ and assign a cost of 1 to each link.
10: Find the shortest path $\{P^*_{u\to v}\}$ between all nodes $u, v \in \mathcal{N}$ in graph $G''(\mathcal{N}, \mathcal{L}'')$.
11: Make a graph $G''''(\mathcal{N}, \mathcal{L}'''')$, where $\mathcal{L}'''' = \{(u, v) \mid r(P^*_{u\to v}) \leq \Delta\}$ and assign a cost of 1 to each link.
12: If there is no link between nodes u and v in graph $G''''(\mathcal{N}, \mathcal{L}'''')$, find the shortest path between nodes $u, v \in \mathcal{N}$ and $w \in T_R$ in graph $G''(\mathcal{N}, \mathcal{L}'')$, for which $r(P^*_{u\to w}) \leq \Delta$ and $r(P^*_{v\to w}) \leq \Delta$, and assign a cost of 1 to each link (u, w) and (v, w) in graph $G'''(\mathcal{N}, \mathcal{L}''')$. Find a shortest path $\{P^*_{u\to v}\}$ in graph $G'''(\mathcal{N}, \mathcal{L}''')$. If there is a path, add link (u, v) in $G''''(\mathcal{N}, \mathcal{L}'''')$ and assign a cost of 1 to the link.
13: On graph G'''' repeat steps $3 - 5$ to obtain path $P^2_{s\to d}$.
14: Remove all links that are both in $P^1_{s\to d}$ and $P^2_{s\to d}$ to obtain two link-disjoint paths.
15: Place shared regenerators for each path.

Algorithm SRPB works in the same way as algorithm DRPB while computing the primary path, but there is a difference in the computation of the backup path. In this algorithm, once the primary path is computed its regenerators are placed temporarily in step 7 to make them available for usage by the backup path. In addition to minimizing the number of regenerators by augmenting the original graph as discussed earlier for algorithm DRPB, this approach further reduces regenerators by reusing the possible regenerators of the primary path while computing the backup path. i.e., it forces regenerator sharing between the primary and backup paths of a request in addition to reusing regenerators of the primary path if the backup path come across those nodes.

Once a graph that contains the temporarily placed primary path regenerator nodes is created by connecting the temporarily placed primary regenerators that are within a distance less than or equal to the physical impairment threshold, the next step is creating a new graph, G'''', on which the process of finding the backup path is computed. In step 9 the shortest distance between all nodes is computed using Dijkstra's algorithm in graph G'' and if the distance computed is less than or equal to the physical impairment threshold, a link connecting the nodes will be added in step 11 while constructing graph G'''', i.e., all nodes that are within a distance of less than or equal to the physical impairment threshold in graph G'' will be directly connected. In addition, if the nodes are not within a distance less than or equal to the physical impairment threshold in graph G'', it computes the shortest paths between those nodes and all nodes in T_R. If length of the paths is less than or equal to the physical impairment threshold, the nodes will be directly connected in graph G'''. Once additional links are added to graph G''', the shortest path between those nodes will be computed in the graph, if there is a path, the two nodes will be connected in graph G''''. The idea here is that, two nodes that are far apart with a distance greater than the physical impairment threshold in the original graph can use the temporarily placed primary regenerators to reach each other, if the temporarily placed regenerators are within a distance of less than or equal to the physical impairment threshold between the two nodes. Steps 3–5 are repeated in graph G'''' to obtain $P^2_{s\to d}$. Step 14 removes the interlacing links between paths $P^1_{s\to d}$ and $P^2_{s\to d}$. Once link-disjoint primary and backup paths are found, step 15 places shared regenerators for the paths.

Complexity of SRPB

The major operations that determine the complexity of algorithm SRPB are: constructing graphs G', G'''' and G''', finding the shortest paths between s and d in these graphs and removing the loops of the paths in G. Finding the shortest paths between each pair of nodes in step 1 has a complexity of $O(N^2 \log N + NL)$. Finding the shortest path between the source and destination nodes in step 3 has a complexity of $O(N^2)$. As discussed earlier, the total complexity of removing loops in step 5 is $O(N^2)$, while the operation in step 8 has a complexity of $O(T_R^2 \log T_R + T_R L)$. Step 10 has a complexity of $O(N^2 \log N + NL)$. In step 12, finding the shortest paths between each pair of nodes and the temporarily placed primary regenerators has the worst complexity of

$O(N^2 \log N + NL)$, while finding the shortest path between those pair of nodes in graph G''' has a complexity of $O(N^2)$. Step 13 repeats steps 3–5 which has a total complexity of $O(N^2)$. Therefore, the total complexity of algorithm SRPB becomes $O(N^2 \log N + NL)$.

2-2-4 Simulation Results

This sub-section discusses the simulation results of the algorithms discussed earlier for the shared survivable regenerator placement problem. The goal of the simulation is to see the performance of the algorithms and make a comparison between them while varying the physical impairment threshold.

The simulation results for an ARPANET network are shown in Figure 2-6. The performance of the algorithms is measured by varying the physical impairment threshold. The algorithms show the same characteristics as discussed earlier for the dedicated survivable regenerator placement problem. The plot shows that the exact algorithm outperforms the heuristic algorithm.

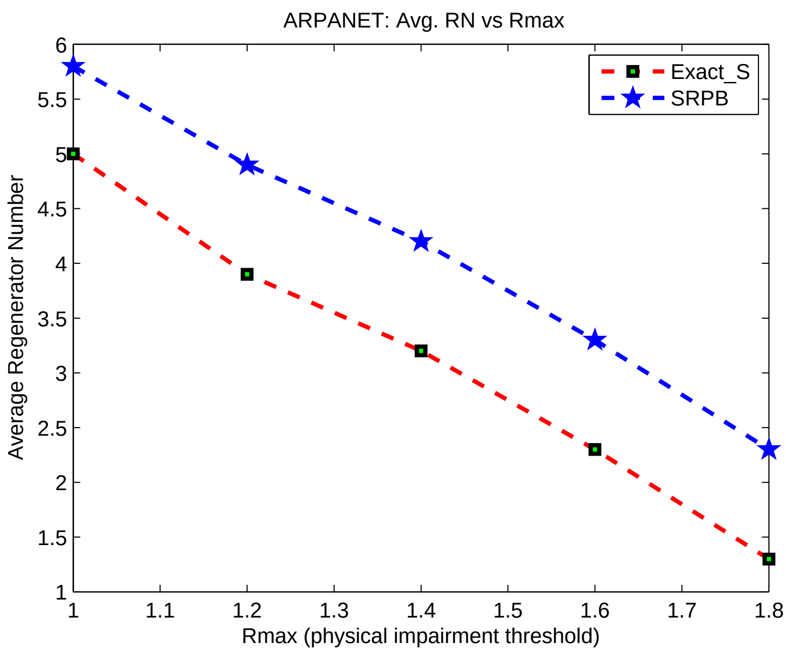

Figure 2-6: Average regenerator number vs physical impairment threshold for the ARPANET network (Exact_S vs SRPB).

2-3 Backup Shared Survivable Regenerator Placement Problem

Another method of minimizing resource usage in a network involves sharing of resources between backup lightpaths. In this sub-problem, regenerators are shared between protection paths of different requests if their working paths are link-disjoint with each other, i.e., as long as their primary paths do not fail simultaneously. Since we are considering only single link failures, when primary paths of different lightpaths are link-disjoint they will not fail at the same time which makes possible to sharing of regenerators between the backup lightpaths.

2-3-1 Problem Definition

1. **Given:** A network topology which is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of N nodes and $\mathcal{L}$ is the set of L links between the nodes; associated with each link $(u, v) \in \mathcal{L}$ there is a wavelength and and a set of non-negative physical impairment metrics $r_i(u, v)$ for $i = 1, \ldots, m$. A lightpath request j is represented by the tuple (s_j, d_j, Δ), where $s_j, d_j \in \mathcal{N}$ are the source and destination nodes of request j and Δ is set of threshold values of the physical impairments.

2. **Objective:** Find a pair of link-disjoint simple paths from the source node to the destination node and place regenerators where backup paths can share regenerators, minimizing the total number of regenerators used by the two paths.

Although solving the ILP problems is still possible for some small sized networks, it is not practical for the problems in real world, large sized networks. The ILP formulations presented earlier for the dedicated and shared survivable regenerator placement problems are very complex. Adding additional formulations to solve the backup shared survivable regenerator placement problem will complicate the algorithm even further, which makes it inconvenient solution. Therefore, to solve the backup shared survivable regenerator placement problem we provide only a heuristic algorithm, SRBB (shared regenerator between backups). The algorithm is given below.

2-3-2 Single Physical Impairment Heuristic Algorithm

To solve this problem, we employ an active path first (APF) or find remove (FR) approach where the primary path is computed first and then its links are dropped before the backup path is computed. This approach is chosen because it is easier to determine the sharing of resources among backup paths when the primary paths are already known.

Algorithm 3 SRBB(G, s, d, Δ)

For each request i,

1: Find the shortest path $\{P^*_{u\to v}\}$ between all nodes $u, v \in \mathcal{N}$, for which $r(P^*_{u\to v}) \leq \Delta$.
2: Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' = \{(u,v) \mid r(P^*_{u\to v}) \leq \Delta\}$ and assign a cost of 1 to each link.
3: Find the shortest path $P'_{s_i\to d_i}$ from s_i to d_i in G'.
4: Substitute the links of $P'_{s_i\to d_i}$ with the corresponding sub-paths $P^*_{s\to d}$ in G to obtain $P_{s\to d}$.
5: Remove all the loops of $P_{s\to d}$ in graph G to obtain path $P^1_{s_i\to d_i}$.
6: Place the necessary regenerators for $P^1_{s\to d}$.
7: Remove all links in $P^1_{s_i\to d_i}$ to obtain $G''(\mathcal{N}, \mathcal{L}'')$.
8: For each primary path that does not share a link with $P^1_{s_i\to d_i}$, set the cost of each link incident to the regenerator nodes of its backup path to zero in graph G''.
9: Repeat steps $1-5$ on G'' to obtain path $P^2_{s_i\to d_i}$.
10: Place the necessary regenerators for $P^2_{s_i\to d_i}$.

Like the other heuristic algorithms discussed earlier, algorithm SRBB starts by connecting all nodes that are within a distance of less than or equal to the physical impairment threshold in graph G due to the same reasons given in previous sections. This process creates a new graph, graph G', on which the primary path computation is performed. In step 3, the shortest path between the source and destination is computed in graph G'. Since graph G' is a sub-graph of graph G, links of the path computed in step 3 are replaced by the corresponding sub-paths in G. However, this path may not be a loopless path since it is made up of a concatenation of path segments. Hence, its loops are removed in step 5 and regenerators are placed according to the requirement of the computed path.

Once the primary path, $P^1_{s\to d}$, is found, the next step is finding the backup path that is link-disjoint with the primary path. The links along the loopless path $P^1_{s\to d}$ are dropped to obtain graph G''. In step 8, the cost of links incident to the regenerator nodes of backup paths of primary paths that are link-disjoint with $P^1_{s\to d}$ are set to zero in graph G'' to encourage regenerator sharing between backup lightpaths. Steps 1–5 are repeated in graph G'' to obtain $P^2_{s\to d}$. If a backup path is found, the algorithm finally places the necessary regenerators for $P^2_{s\to d}$.

Even though the find remove method has a disadvantage of not finding a link-disjoint path, even if there are, due to its link removal operation, it helps minimizing regenerators used by the backup path. Once a primary path is computed, the algorithm checks whether this primary path is link-disjoint with primary paths of previously admitted requests. After dropping links of the primary path from the graph, it encourages regenerator sharing between backup paths by setting the cost of each link incident to the regenerator nodes of backup paths to zero. Thus, it facilitates regenerator minimization used by a backup path.

Complexity of SRBB

As described earlier for the other algorithms, operations in steps 1 and 2 have a complexity of $O(N^2 \log N + NL)$. Finding the shortest path in step 3 has a complexity of $O(N^2)$. Removing the loops in step 5 has a complexity of $O(N^2)$, while the operation in step 8 has a complexity of $O(lN^2)$, where l is the number of requests. And finally, repeating the operations 1–5 in step 9 for computing the backup path will have the same complexity as discussed for the primary path. Therefore, the total complexity of algorithm SRBB is $O(N^2 \log N + lN^2 + NL)$.

2-3-3 Simulation Results

The simulation environment used for this simulation is the same as we have been using in previous simulations. But, unlike the previously discussed simulations, multiple requests are considered in this simulation since the idea is to share regenerators between backup paths of different requests.

The simulation results are shown on the graphs Figure 2-7–Figure 2-9. Since we have not provided an ILP formulation for this sub-problem, we will compare the results of algorithm SRBB against the DRPB algorithm given earlier. The performance of the algorithm is measured by varying the physical impairment threshold. The algorithm shows the same behavior as previously discussed algorithms when varying the physical impairment threshold.

The graphs shown in Figure 2-7–Figure 2-9 show the simulation results of DRPB vs SRBB for 100 flows and 10 iterations. As can be seen from the graphs, we can reduce the amount of regenerators required by a set of requests by sharing regenerators among backup paths whose primary paths are link-disjoint, i.e., the primary paths do not fail simultaneously.

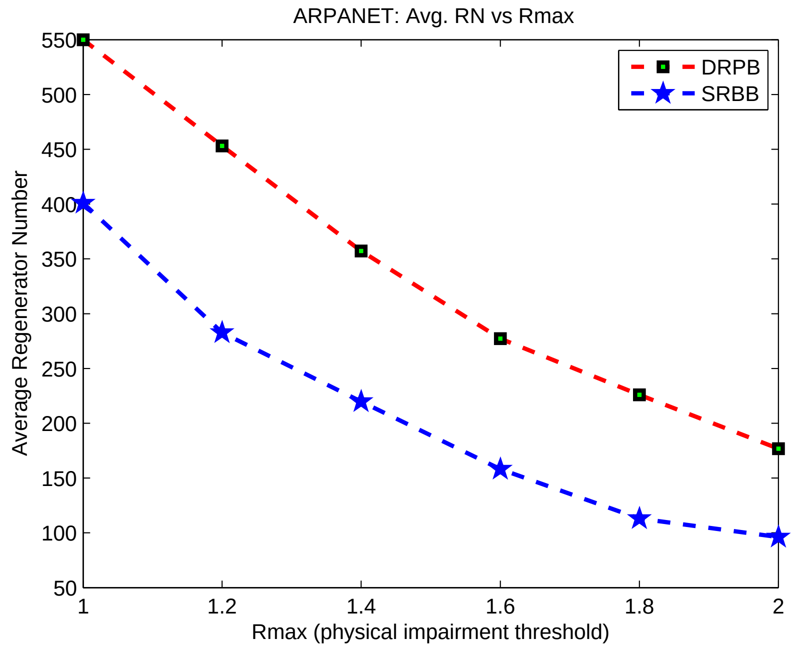

Figure 2-7: Average regenerator number vs physical impairment threshold for the ARPANET network (DRPB vs SRBB).

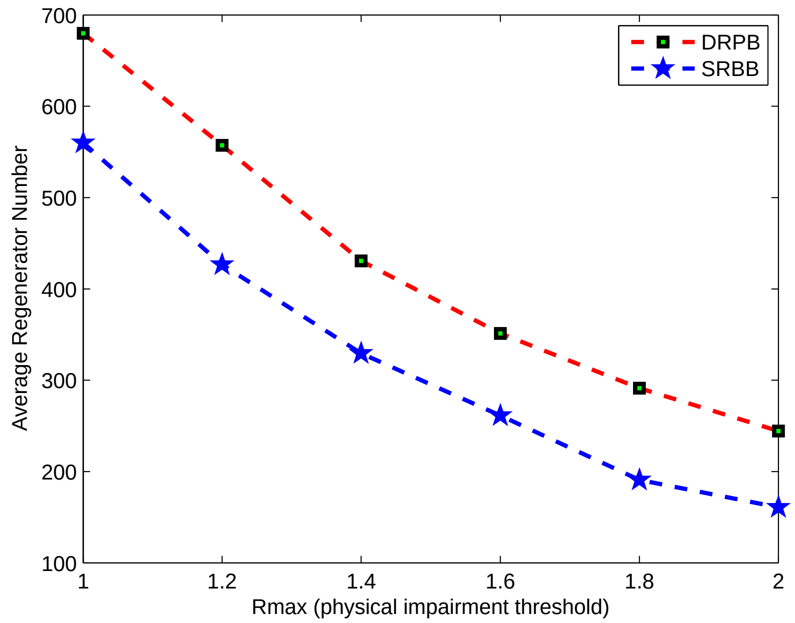

Figure 2-8: Average regenerator number vs physical impairment threshold for lattice network with $N = 49$ (DRPB vs SRBB).

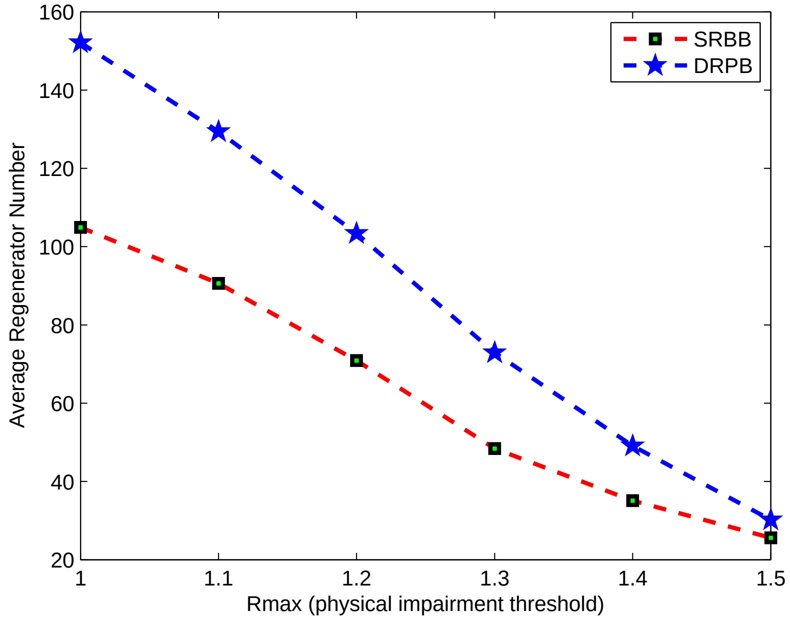

Figure 2-9: Average regenerator number vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (DRPB vs SRBB).

2-4 Multiple Physical Impairments Heuristic Algorithms

To solve the survivable regenerator placement problem for multiple physical impairments ($i = 1, \ldots, m$), we used extension of the three heuristic algorithms given earlier for a single physical impairment to multiple dimensions in addition to the exact ILP algorithm which we extend to support multiple physical impairments. MPI_Exact is an exact algorithm in which assignment of regenerators is dedicated for the primary and the backup path of a request, while MPI_Exact_S is an exact algorithm that handles regenerator sharing between primary and backup paths of the same request.

The heuristic algorithms are MPI_DRPB, MPI_SRPB and MPI_SRBB. MPI_DRPB algorithm assigns dedicated regenerators for the primary and backup paths, MPI_SRPB is an algorithm in which regenerators are shared between the primary and backup path of the same request, whereas algorithm MPI_SRBB allows sharing of regenerators between backup paths of different connections whose primary paths are link-disjoint.

2-4-1 Dedicated Survivable Regenerator Placement

In this sub-problem dedicated regenerators are placed for the primary and backup paths of a request.

Algorithm 4 MPI_DRPB(G, m, s, d, Δ)

1: Find the shortest path $\{P^*_{u\to v}\}$ between all nodes $u, v \in \mathcal{N}$ for the first metric, for which $r_i(P^*_{u\to v}) \leq \Delta$ for $1 \leq i \leq m$, using Dijkstra's algorithm.
2: Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' = \{(u, v) \mid r_i(P^*_{u\to v}) \leq \Delta\}$ for $1 \leq i \leq m$ and assign a cost of 1 to each link.
3: Find the shortest path $P'_{s\to d}$ from s to d in G' using Dijkstra's algorithm.
4: Substitute the links of $P'_{s\to d}$ with the corresponding sub-paths $P^*_{u\to v}$ in G to obtain $P_{s\to d}$.
5: Remove all the loops of $P_{s\to d}$ in graph G to obtain path $P^1_{s\to d}$.
6: Redirect all links in $P^1_{s\to d}$ from d to s to obtain $G''(\mathcal{N}, \mathcal{L}'')$.
7: On graph G'' repeat steps $1 - 5$ to obtain path $P^2_{s\to d}$.
8: Remove all links that are both in $P^1_{s\to d}$ and $P^2_{s\to d}$ to obtain two link-disjoint paths.
9: Place dedicated regenerators for each path that is feasible for all the physical impairment metrics.

Algorithm MPI_DRPB works in the same way as algorithm DRPB except in steps 1 and 2. In step 1, the shortest path between all nodes is computed using Dijkstra's algorithm in graph G for one of the physical impairments and if the distance computed in step 1 is less than or equal to the physical impairment threshold for all the impairments, a link connecting the nodes will be added in step 2 while constructing graph G'. After construction of graph G', the algorithm performs in the same way as described earlier for algorithm DRPB.

Complexity of MPI_DRPB

Since most of the meta code of MPI_DRPB is the same as the meta code of DRPB, it will only have one additional component in the total complexity. The construction of the graphs involves finding the shortest paths between each pair of nodes. This can be implemented with complexity of $O(N^2 \log N + NL)$ for one metric. The checking of feasibilities of the shortest paths for the other remaining metrics has a complexity of $O(m)$. Finding the shortest paths in G' and G'' will have a complexity of $O(N^2)$. The complexity of removing loops is $O(N^2)$. Therefore, the total complexity of algorithm MPI_DRPB is $O(N^2 \log N + NL + m)$.

2-4-2 Shared Survivable Regenerator Placement

Algorithm 5 MPI_SRPB(G, m, s, d, Δ)

1: Find the shortest path $\{P^*_{u \to v}\}$ between all nodes $u, v \in \mathcal{N}$ for the first metric, for which $r_i(P^*_{u \to v}) \leq \Delta$ for $1 \leq i \leq m$, using Dijkstra's algorithm.
2: Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' = \{(u, v) \mid r_i(P^*_{u \to v}) \leq \Delta\}$ for $1 \leq i \leq m$ and assign a cost of 1 to each link.
3: Find the shortest path $P'_{s \to d}$ from s to d in G'.
4: Substitute the links of $P'_{s \to d}$ with the corresponding sub-paths $P^*_{u \to v}$ in G to obtain $P_{s \to d}$.
5: Remove all the loops of $P_{s \to d}$ in graph G to obtain path $P^1_{s \to d}$.
6: Redirect all links in $P^1_{s \to d}$ from d to s to obtain $G''(\mathcal{N}, \mathcal{L}'')$.
7: Temporarily place regenerators, T_R, for path $P^1_{s \to d}$.
8: Find the shortest path $\{P^*_{u \to v}\}$ between all nodes $u, v \in T_R$ in graph $G''(\mathcal{N}, \mathcal{L}'')$.
9: Make a graph $G'''(\mathcal{N}, \mathcal{L}''')$, where $\mathcal{L}''' = \{(u, v) \mid r(P^*_{u \to v}) \leq \Delta\}$ and assign a cost of 1 to each link.
10: Find the shortest path $\{P^*_{u \to v}\}$ between all nodes $u, v \in \mathcal{N}$ in graph $G''(\mathcal{N}, \mathcal{L}'')$.
11: Make a graph $G''''(\mathcal{N}, \mathcal{L}'''')$, where $\mathcal{L}'''' = \{(u, v) \mid r(P^*_{u \to v}) \leq \Delta\}$ and assign a cost of 1 to each link.
12: If there is no link between nodes u and v in graph $G''''(\mathcal{N}, \mathcal{L}'''')$, find the shortest path between nodes $u, v \in \mathcal{N}$ and $w \in T_R$ in graph $G''(\mathcal{N}, \mathcal{L}'')$, for which $r(P^*_{u \to w}) \leq \Delta$ and $r(P^*_{v \to w}) \leq \Delta$, and assign a cost of 1 to each link (u, w) and (v, w) in graph $G'''(\mathcal{N}, \mathcal{L}''')$. Find a shortest path $\{P^*_{u \to v}\}$ in graph $G'''(\mathcal{N}, \mathcal{L}''')$. If there is a path add link (u, v) in $G''''(\mathcal{N}, \mathcal{L}'''')$ and assign a cost of 1 to the link.
13: On graph G'''' repeat steps 3 – 5 to obtain path $P^2_{s \to d}$.
14: Remove all links that are both in $P^1_{s \to d}$ and $P^2_{s \to d}$ to obtain two link-disjoint paths.
15: Place shared regenerators for each path.

As described earlier, regenerators are shared betweeen primary and backup paths of the same request in this sub-problem. Algorithm MPI_SRPB works in the same way as algorithm SRPB except in its initial two steps. All nodes that are within a distance of less than or equal to the physical impairment threshold for all the metrics will be directly connected to create new graph G'. After construction of graph G', algorithm MPI_SRPB performs in the same way as algorithm SRPB.

Complexity of MPI_SRPB

As described for algorithm SRPB, finding the shortest paths between each pair of nodes in step 1 has a complexity of $O(N^2 \log N + NL)$ for one metric. While checking feasibilities of the shortest paths for the other remaining metrics has a complexity of $O(m)$. Finding the shortest path between the source and destination in step 3 has a complexity of $O(N^2)$. The complexity of removing loops in step 5 is $O(N^2)$, while complexity of the operations in steps 8 and 9 will have a complexity of $O(T_R^2 \log T_R + T_R L)$. Step 10 has a complexity of $O(N^2 \log N + NL)$. As shown for algorithm SRPB, step 12 has the worst complexity of $O(N^2 \log N + NL)$. Step 13 repeats steps 3–5 which has a total complexity of $O(N^2)$. Therefore, the total complexity of algorithm MPI_SRPB becomes $O(N^2 \log N + NL + m)$.

2-4-3 Backup Shared Survivable Regenerator Placement

In this sub-problem, backup paths of different requests share regenerators as long as their primary paths are link-disjoint.

Algorithm 6 MPI_SRBB(G, m, s, d, Δ)

For each request i,

1: Find the shortest path $\{P^*_{u\to v}\}$ between all nodes $u, v \in \mathcal{N}$ for the first metric, for which $r_j(P^*_{u\to v}) \leq \Delta$ for $1 \leq j \leq m$, using Dijkstra's algorithm.
2: Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $\mathcal{L}' = \{(u, v) \mid r_j(P^*_{u\to v}) \leq \Delta\}$ for $1 \leq j \leq m$ and assign a cost of 1 to each link.
3: Find the shortest path $P'_{s_i\to d_i}$ from s_i to d_i in G' using Dijkstra's algorithm.
4: Substitute the links of $P'_{s_i\to d_i}$ with the corresponding sub-paths $P^*_{s\to d}$ in G to obtain $P_{s\to d}$.
5: Remove all the loops of $P_{s\to d}$ in graph G to obtain path $P^1_{s_i\to d_i}$. Place the necessary regenerators for $P^1_{s\to d}$.
6: Remove all links in $P^1_{s_i\to d_i}$ to obtain $G''(\mathcal{N}, \mathcal{L}'')$.
7: For each primary path that does not share a link with $P^1_{s_i\to d_i}$, set the cost of each link incident to the regenerator nodes of its backup path to zero.
8: Repeat steps 1–5 to obtain path $P^2_{s_i\to d_i}$. Place the necessary regenerators for $P^2_{s_i\to d_i}$.

Algorithm MPI_SRBB works in the same way as algorithm SRBB except in steps 1 and

2. In step 1 shortest distance between all nodes is computed using Dijkstra's algorithm in graph G for one of the impairments and if the distance computed in step 1 is less than or equal to the impairment threshold for all the impairments, a link connecting the nodes will be added in step 2 while constructing graph G'. After creation of graph G' by connecting all nodes that are within a distance of less than or equal to the physical impairment thresholds of all the metrics, the algorithm performs the path computation process as described earlier for algorithm SRBB.

Complexity of MPI_SRBB

The operations in steps 1 and 2 have a complexity of $O(N^2 \log N + NL + m)$. Finding the shortest path between the source and destination of a request has a complexity of $O(N^2)$. While removing the loops of a path in step 5 has a complexity of $O(N^2)$, the operation in step 8 has a complexity of $O(lN^2)$, where l is the number of requests. Therefore, the total complexity of algorithm MPI_SRBB is $O(N^2 \log N + lN^2 + NL + m)$.

2-4-4 Simulation Results

The simulations of the algorithms proposed for the problem of multiple impairments-aware survivable regenerator placement problem are performed for $m = 2$ with the same threshold values and are shown in this sub-section.

Figure 2-10 and Figure 2-11 show that the heuristic algorithms, MPI_DRPB and MPI_SRPB, require more regenerators than the exact algorithms, MPI_Exact and MPI_Exact_S, for the dedicated and shared scenarios respectively. The figures also show that, as the physical impairment threshold increases the number of regenerators required decreases due to the same reason discussed earlier for the other algorithms.

Figure 2-12 and Figure 2-13 show that the Exact algorithms, MPI_Exact and MPI_Exact_S, have a larger running time compared to MPI_DRPB and MPI_SRPB algorithms respectively. The exact algorithms have larger running time since they are ILP formulations which have exponential complexity. In Figure 2-12, the running time of the exact algorithm increases with the physical impairment threshold. When the physical impairment threshold increases, more paths will not violate the physical constraint which leads to longer running time. In Figure 2-13, the running time of the exact algorithm increases to some point then starts to decrease. In this scenario, when the physical impairment threshold is medium we have a longer running time. This result arises because, in this region a path can be easily found compared to when the physical constraint is small which leads to the increase of variables in the ILP formulation.

Figure 2-14 and Figure 2-15 show that for the ARPANET network, the dedicated

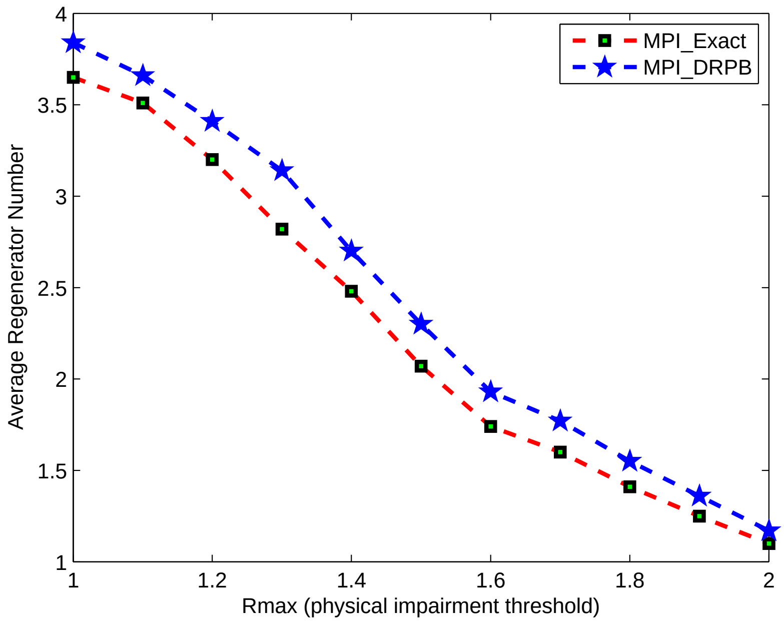

Figure 2-10: Average regenerator number vs physical impairment threshold for NSFNET network (MPI_Exact vs MPI_DRPB).

algorithm, MPI_DRPB, requires more regenerators than the shared ones, MPI_SRPB and MPI_SRBB. This shows that, we can reduce the cost of a network by sharing regenerators between primary and backup paths of the same request as far as they do not need it simultaneously and between backup paths whose primary paths are link-disjoint. The results of the simulations conducted on random and lattice networks also show similar behaviors as the ARPANET network, and are presented in Appendix A-1.

The running time vs physical impairment threshold plots of the multiple physical impairments algorithms for the ARPANET network are shown in Figure 2-16 and Figure 2-17. The small fluctuation in the running time plot of Figure 2-17 results from the difference in the state of the processor when the different data points are taken. At each data point, the standard deviation is computed and it shows that most of the data points are found to be in the order of the average value.

The figures show that the dedicated algorithm (MPI_DRPB) has a lower running time compared to its shared counterparts (MPI_SRPB and MPI_SRBB); this is because, in the case of regenerator sharing between backup paths, while computing the backup path the algorithm checks disjointedness of primary paths of admitted flows before the current flow with the current flow's primary path and makes a new graph as shown in the meta code so that their backup paths can share a regenerator. In case of regenerator sharing between primary and backup paths of the same request, when computing the backup path

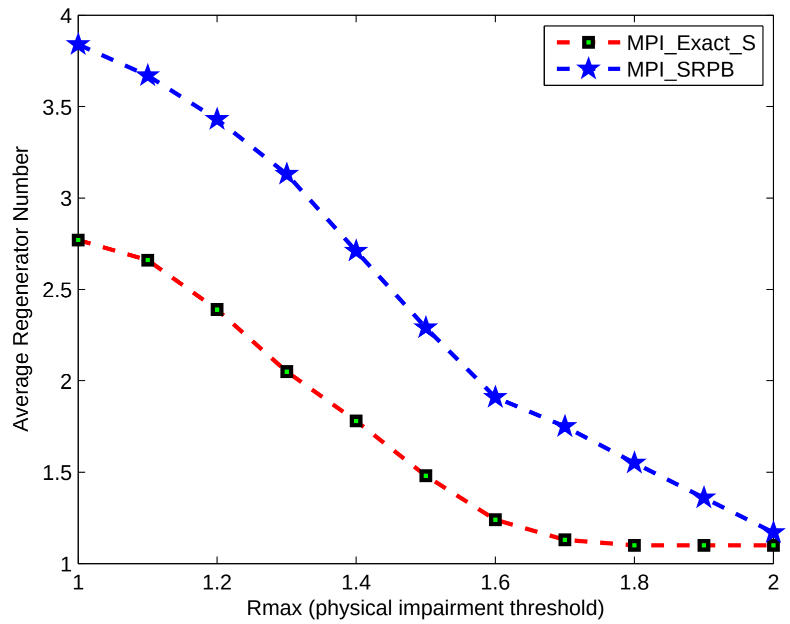

Figure 2-11: Average regenerator number vs physical impairment threshold for NSFNET network (MPI_Exact_S vs MPI_SRPB).

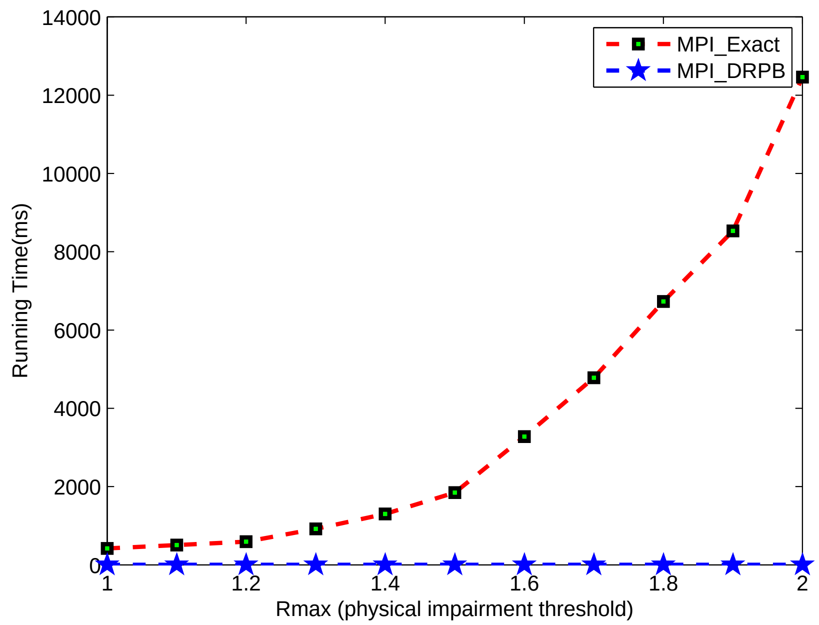

Figure 2-12: Running time vs physical impairment threshold for NSFNET network (MPI_Exact vs MPI_DRPB).

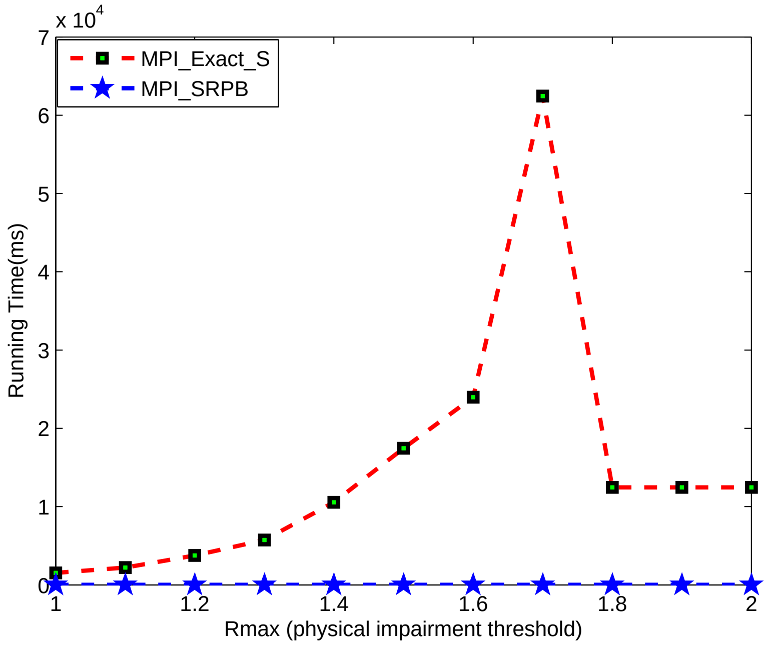

Figure 2-13: Running time vs physical impairment threshold for NSFNET network (MPI_Exact_S vs MPI_SRPB).

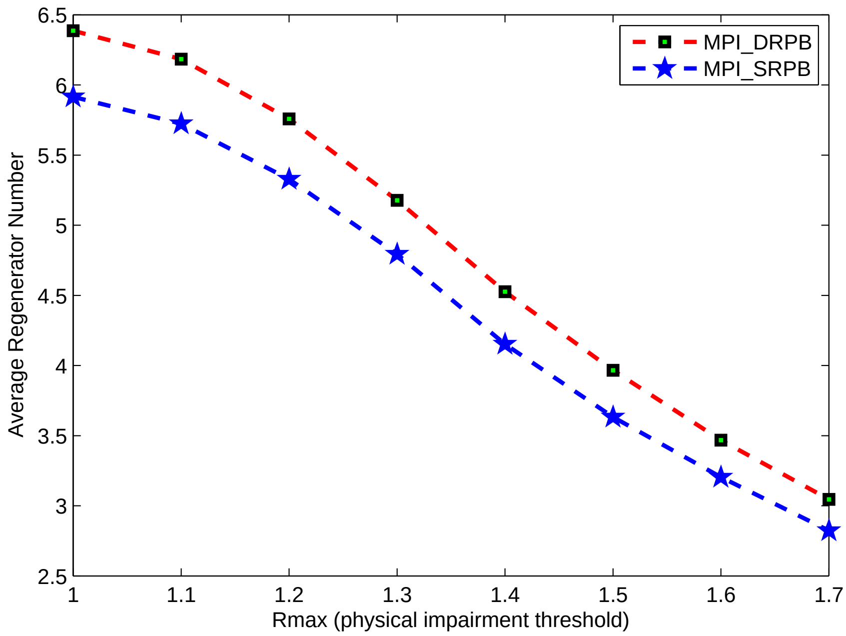

Figure 2-14: Average regenerator number vs physical impairment threshold for the ARPANET network (MPI_DRPB vs MPI_SRPB).

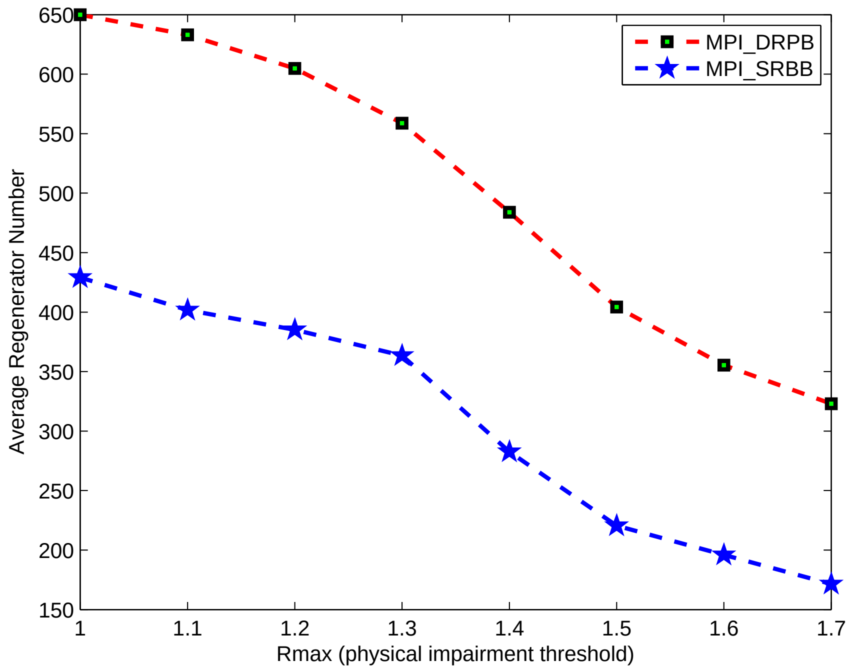

Figure 2-15: Average regenerator number vs physical impairment threshold for the ARPANET network (MPI_DRPB vs MPI_SRBB).

the algorithm builds a sub-graph that consists of the possible regenerator nodes of the primary path and extra work is done while performing the regenerator placement of both paths compared to its dedicated counterpart.

In addition, the figures depict that as the physical impairment threshold increases, the running time of MPI_SRPB decreases. This result arises due to the fact that as the physical impairment threshold increases the number of regenerators temporarily placed for the primary path decreases, which reduces the running time of building the graph containing those nodes and computing a path between pair of nodes in that graph.

The simulations performed on lattice and random networks also show similar patterns, and are provided in Figure A-5–Figure A-8.

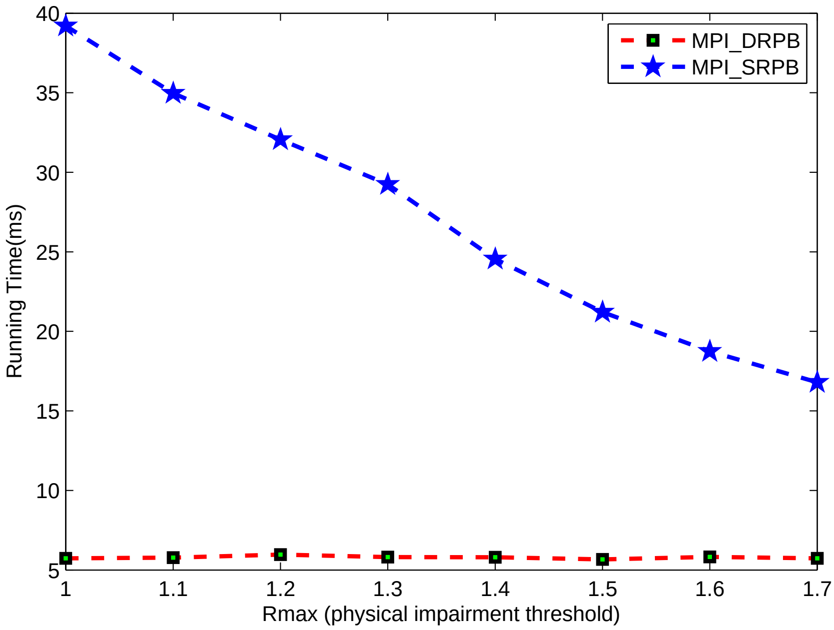

Figure 2-16: Running time vs physical impairment threshold for the ARPANET network (MPI_DRPB vs MPI_SRPB).

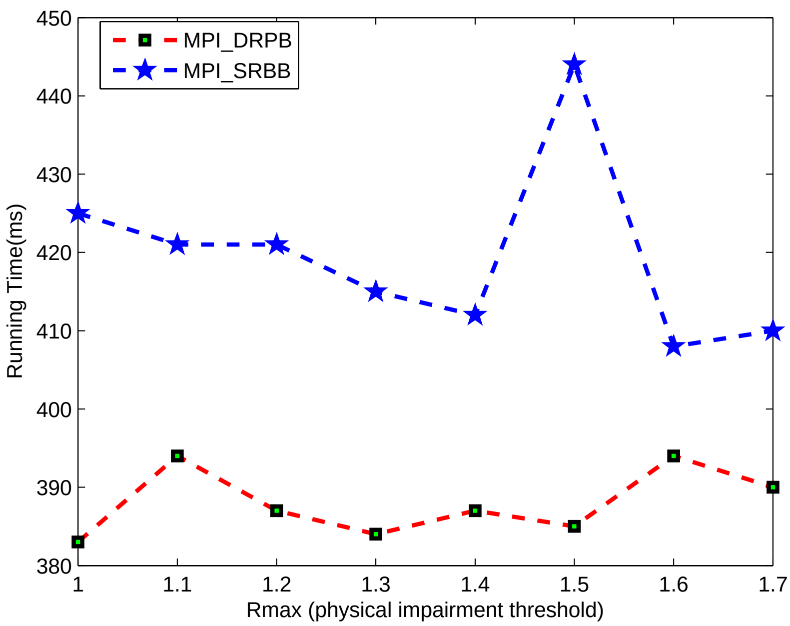

Figure 2-17: Running time vs physical impairment threshold for the ARPANET network (MPI_DRPB vs MPI_SRBB).

Chapter 3

Impairment-aware Survivable Routing Problem

Finding a survivable path in translucent optical networks, networks with sparsely placed regenerators, is the other problem we study in this thesis.

This problem is divided into different sub-problems: dedicated impairment-aware survivable routing problem, shared impairment-aware survivable routing problem, backup shared impairment-aware survivable routing problem, dedicated impairment-aware survivable routing problem under dynamic traffic and backup regenerator and wavelength shared impairment-aware survivable routing problem. The definitions and solutions of each sub-problems are given in subsequent sections.

3-1 Dedicated Impairment-aware Survivable Routing Problem

In this sub-problem, the primary and backup lightpaths of a request uses dedicated regenerators. As discussed in the previous chapter, dedicated assignment of regenerators is used in a scenario in which the network is protected by the 1+1 protection mechanism where both the primary and backup lightpaths are transmitting a signal and may require regeneration at the same time. Since we are considering dedicated protection, the different requests do not share regenerators. Thus, each request can be considered individually. i.e., the problem is reduced to a single request problem.

Notations:
The notations used in the upcoming algorithms are given below.

- N_R is the number of regenerator nodes in the network.
- m is the number of the physical impairments associated with each fiber link.
- $\overrightarrow{q}(u,v)$ is the physical impairments vector associated with the fiber link (u,v), where $\overrightarrow{q}_i(u,v)$, $1 \le i \le m$, is the i^{th} physical impairment on the fiber link.
- $\overrightarrow{\Delta}$ represents the physical impairment threshold vector, where Δ_i represents the threshold value of the i^{th} physical impairment ($1 \le i \le m$).
- $\overrightarrow{I}(p)$, is a vector of sums, where $I_i(p)$ represents the sum of the i^{th} physical impairment along the path p since the last regeneration (or since the source node if there was no regeneration).
- $\overrightarrow{I}^*(p)$, is a sum vector, where $I_i(p)$ denotes the sum of the i^{th} physical impairment along the path p since the last regenerator node (or since the source node if no regenerator node is encountered).
- $Adj[u]$ represents the set of nodes that are adjacent to node u in the graph G.
- $\pi[u[i]]$ represents the set of nodes that appear in the i^{th} path stored at node u.
- $lur(u[i])$ represents the last unused (free) regenerator along the i^{th} path stored at node u.
- $reg\_numb(u[i])$ is the number of regenerators used in the i^{th} path stored at node u.
- R is a set of nodes including all the regenerator nodes in the network and the destination node of the request.
- $p_{n\rightarrow j;i}$ represents the shortest path from node n to the nearest node j, $j \in R$, with respect to the i^{th} physical impairment.
- $\overrightarrow{b}(n)$ denotes the attainable lower bounds on the distance to the nearest node $j \in R$ for each physical impairment, where $\overrightarrow{b}_i(n)$ is the accumulated sum of the i^{th} physical impairment on the path $p_{n\rightarrow j;i}$.
- $maxlen$ refers to the maximum length of the physical impairment.

3-1-1 Problem Definition

1. **Given:** A network topology which is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of N nodes and $\mathcal{L}$ is the set of L links between the nodes; associated with each link $(u, v) \in \mathcal{L}$ there is a wavelength and a set of non-negative physical impairment metrics $r_i(u, v)$ for $i = 1, \ldots, m$. $\mathcal{N}_R \subseteq \mathcal{N}$ represents the set of R nodes that have regeneration capability, and $N_R = |\mathcal{N}_R|$ represents the number of regenerator nodes. A lightpath request is represented by the tuple (s, d, Δ), where $s, d \in \mathcal{N}$ are the source and destination nodes of the request and Δ is the set of threshold values of the physical impairments.

2. **Objective:** Find feasible link-disjoint primary and backup simple paths, that use dedicated regenerators, from the source node to the destination node, i.e., the two paths use dedicated regenerators and do not violate the physical impairment constraint.

We formulate the problem of finding dedicated link-disjoint primary and backup paths in presence of scarce regenerator nodes in the network using both an exact approach and a heuristic approach. The algorithms, D_EISRA, D_H1ISRA and D_H2ISRA, used to solve the problem are given below.

3-1-2 Exact Algorithm

The problem will be solved using an exact algorithm, D_EISRA, that uses some concepts of the self-adaptive multi-constrained routing algorithm (SAMCRA) [46]. SAMCRA is an exact multi-constrained shortest path algorithm which finds an optimal path, if there exists any, that satisfies all the constraints. Before we proceed to explain how SAMCRA is modified to solve the impairment-aware survivable routing problem, a brief description of SAMCRA is given below.

SAMCRA mainly utilizes four concepts:

- **A non-linear path length:** Path length is a means to compare different (sub)paths with different impairment metrics. In SAMCRA, the path length of a given path p, represented as $l(p)$, is a non-linear function of the link weights.

$$l(p) = \max_{1 \leq i \leq m} \left[\frac{I_i(p)}{\Delta_i} \right] \tag{3-1}$$

- **k-shortest paths:** Unlike Dijkstra's shortest path algorithm, which stores at each intermediate node the previous hop and the length of the shortest path from the source node to the intermediate node, SAMCRA stores up to k shortest paths and their corresponding lengths.

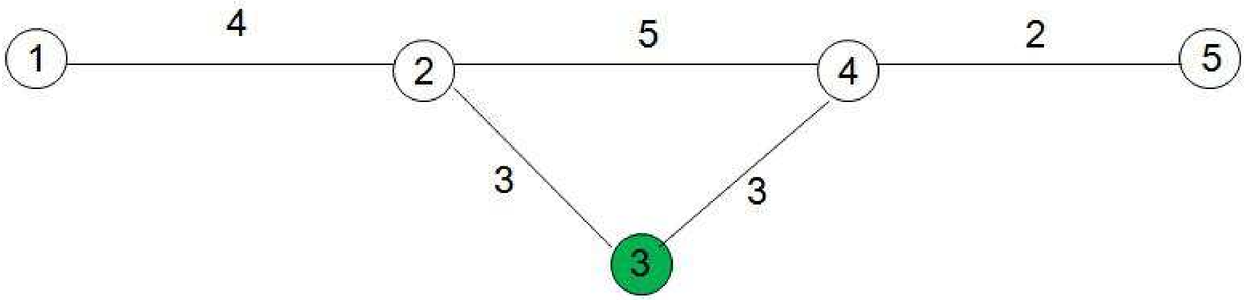

Figure 3-1: An example showing that the dominance concept fails for impairment-aware routing.

- **Non-dominated paths:** At any intermediate node, it does not make sense to store a path that has worse impairment metrics (i.e., higher or equal in every metric and exactly higher in at least one) than another path. Such paths are said to be dominated and are discarded, this reduces the search space for possible paths thereby increasing the computing efficiency.

- **Look-ahead:** The purpose of the look-ahead concept, just like the idea of non-dominated paths, is to reduce the search space for possible paths. Look-ahead utilizes information related to the remaining sub-path towards the destination in order to predict whether the current sub-path can possibly exceed any of the impairment constraints. The information is built by computing (for each metric) the shortest path tree rooted at the destination to each node in the network.

While the look-ahead concept can be applied with some modifications, the non-dominance technique fails in impairment-aware routing due to regeneration of a signal at a regenerator node where the value of the physical impairment is reset to zero. The network shown in Figure 3-1 is an example where the dominance concept fails.

In this example, we are assuming that $m = 1$, the connection request is $(1, 5, 10)$ and only node 3 has regeneration capability. At node 4, the sub-path $p_1 = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ with $I(p_1) = 10$ and $I^*(p_1) = 3$ is dominated by the sub-path $p_2 = 1 \rightarrow 2 \rightarrow 4$ with $I(p_2) = I^*(p_2) = 9$. If we drop p_1 due to it being dominated by p_2, a solution can not be obtained for the current request. However, if we keep p_1, it will be part of the only feasible path $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$.

Since SAMCRA can not be directly used to solve the impairment-aware survivable routing problem due to its incapability to handle regeneration and failure of the non-dominance technique due to regeneration, we have built an exact algorithm that uses some of the concepts of SAMCRA and also incorporates additional features to handle regeneration. The non-linear definition of path length and the k-shortest paths concepts of SAMCRA are adopted in the new algorithm.

Let R be the set including all nodes with regeneration capability and the destination node. Let $p_{n \to b;i}$ be the shortest path from node n to the nearest node b ($b \in R$) with respect to the i^{th} physical impairment. The vector $\overrightarrow{b}(n)$ with $b_i(n) = I_i^*(p_{n \to b;i})$ represents the attainable lower bounds on the distance to the nearest regenerator node or the destination node for each physical impairment. At each node n, we compute the look-ahead sum $I_i(p_{s \to n}) + b_i(n)$ for each $i = 1, 2, \ldots, m$. This sum represents the sum of the i^{th} physical impairment between the last regenerator node and the next nearest regenerator node. If the sum exceeds the threshold value Δ_i, then the (sub)path is discarded. Because the non-dominance test automatically avoids loops while computing a path, there is no need of explicit checking for occurrence of loops in SAMCRA. Since the non-dominance technique fails in impairment-aware routing, presence of loops is explicitly checked in the modified algorithm.

Algorithm 7 $D\_EISRA(G, m, s, d)$

1: Clone graph G (node i becomes node $i + N$) to graph G'
2: Connect the two graphs, G and G', via a link between node d and node $s + N$ with weight of Δ and make graph G''.
3: $M\_SAMCRA(G'', m, s, d')$ {find a path between s and $d' = d + N$ in graph G''}

The exact algorithm, D_EISRA (Dedicated Exact Impairment-aware Survivable Routing Algorithm), is described as follows. The subroutine D_EISRA creates a new graph on which the main algorithm computes the path. Step 1 creates a duplicate of the original graph by assigning node ID of $i + N$ for the duplicate of node i; the weights of the links remains the same. In step 2 the algorithm connects the two graphs, G and G', via a link between node d and node $s + N$ with weight of Δ which creates graph G''. In simulation, it is assumed that there are regenerators at nodes d and $s + N$. In step 3 it calls the main algorithm, M_SAMCRA (Modified SAMCRA), to compute a path from source (s) to destination ($d + N$).

In M_SAMCRA, steps 1 and 2 set the number of stored paths (*counter*) at each node to zero. If $l^*(p) > maxlen$, we try to regenerate the signal; and if regeneration is not possible, the (sub)path p is discarded. $maxlen$ is set to 1.0 in step 4.

In step 6, the shortest distance $b_i(n)$ from each node $n \in N$ to the nearest node in R is calculated for each individual physical impairment i. This is done by computing, for each physical impairment, the shortest path tree with Dijkstra algorithm from each node in R to all other nodes. In step 10 the last unused regenerator ($last\_reg$) is set to 0. In

Algorithm 8 $M\_SAMCRA(G, m, s, d)$

```
for each v ∈ V do
    counter[v] ← 0
end for
maxlen ← 1.0
for i = 1, ..., m do
    DIJKSTRA(G, {N_R, d}, i) → b_i(n)
end for
queue Z ← 0
counter[s] ← counter[s] + 1
last_reg ← 0
INSERT (Z,num,0.0,l(→b(s)), 0, 0,last_reg, 0, s, counter[s])
while (Z ≠ 0) do
    EXTRACT-MIN(Z) → u[i]
    u[i] ← GREY
    if (u = d) then
        STOP→ return path
    else
        for each v ∈Adj[u] AND v ∉ {π[u[i]], s} do
            flag_ru ← 0
            →pc_slr ← →I*(u[i]) + →q_{u→v}
            →pc ← →I(u[i]) + →q_{u→v}
            if (l(→pc) > maxlen) then
                if (lur(u[i]) ≠ 0) then
                    →pc ← →pc_slr
                    flag_ru ← 1
                    reg(u[i] + N) ← reg(u[i] + N) − 1
                end if
            end if
            if (flag_ru = 1) then
                reg_count ← reg_numb(u[i]) + 1
                last_reg ← 0
            else
                reg_count ← reg_numb(u[i])
                last_reg ← lur(u[i])
            end if
            look_aheadPC ← l(→I*(u[i]) + →q_{u→v} + →b*[v])
            if ((look_aheadPC ≤ maxlen AND (l(→pc) ≤ maxlen) then
                if reg(v) ≠ 0 then
                    last_reg ← v
                    →pc_slr ← →0
                end if
                counter[v]← counter[v]+1
                INSERT(Z,num,prediction,reg_count,l(→pc), last_reg, l(→pc_slr),v,counter[v])
                →I(v[counter[v]]) ← →pc
                lur(v[counter[v]]) ← last_reg
                →I*(v[counter[v]]) ← →pc_slr
                π[v[counter[v]]] ← u[i]
            end if
        end for
    end if
end while
```

step 11 the source node s is inserted into the queue Z.

As long as the the queue Z is not empty (otherwise no feasible path is present), the *extract_min* function in step 13 extracts the minimum path length in the queue. The extracted path $u[i]$ represents the i^{th} path $p_{s \to u}$ stored in the queue at node u. In step 14, the extracted path is marked gray. If the node u corresponding to the extracted path $u[i]$ equals the destination d, the shortest path satisfying the constraints is returned and the algorithm stops (step 16). But, if $u \neq d$, the scanning procedure starts in step 18.

In steps 18-51, the i^{th} path up to node u is extended toward its neighboring node v, except for the previous nodes in the path $u[i]$ or previous nodes that lead to a common link depending on where the node resides. Step 18 explicitly checks by back tracing if v is in the previous nodes of $u[i]$ to prevent looping and link disjointedness is also checked. When node v is in the first part of the graph previous node check is enough. Whereas, when node v is in the second part of the graph, the algorithm checks whether it is a previous node within the second part of the graph or the link connecting node v and the node u is already in the path. i.e., while computing the backup path the algorithm checks the disjointedness of its links with the primary path in addition to checking previous nodes within the second part of the graph.

Steps 20 and 21 compute the physical impairment weight vector($\overrightarrow{pc}$), and the weight vector of the physical impairment since the last regenerator ($\overrightarrow{pc}_{slr}$) of the extended path. Step 22 tests if the physical impairment weight of the extended path violates the physical constraint. If so, it checks if there is an unused regenerator along the sub-path $u[i]$ (step 23); and if one is found the weight vector of the physical impairment is set to the weight vector of the physical impairment beginning from the last unused regenerator (step 24), and the $flag_{ru}$ is set to 1 to remember that a regenerator has just been used. In steps 29 to 35, the values of regenerator count (reg_count) and the last unused regenerator (lur) are updated depending on the value of $flag_{ru}$. If $flag_{ru} = 1$, reg_count is incremented, and lur is set to 0. Otherwise, the corresponding values are copied from $u[i]$.

In step 36, the *look_aheadPC* is calculated, which refers to the length of the predicted physical impairment vector to the nearest node $n \in R$. Step 37 checks if the new extended path satisfies the look_ahead conditions and obeys the physical impairment constraint. If this is the case and if v has an unused regenerator, lur and $\overrightarrow{pc}_{slr}$ are updated accordingly in steps 39–40). In steps 42–43, the *counter* of node v is incremented, and the new extended path is inserted into the queue Z. In steps 44-48, the new extended path is stored at the current *counter* index of v with all its parameters.

Complexity of D_EISRA

The complexity of the exact algorithm is given as follows. We only considered the major operations to determine the complexity. The initialization of the counter in step 2 takes O(N) times. The computation of the lower bounds in step 6 which uses Dijkstra algorithm ($N_R + 1$) times has a time complexity of $mO(N_R(N \log N + L))$. The maximum number of paths that the queue Z can contain is $k_{max}N$. From [46], extracting a minimum path length among $k_{max}N$ different path lengths takes at most $O(\log(k_{max}N))$. Since k_{max} paths can be stored at each node in the queue, the extraction process in step 13 takes $O(k_{max}N \log(k_{max}N))$. Returning the path in step 16 takes at most $O(N)$. Since the for loop starting in step 18 is invoked at most k_{max} times from each side of each link in the graph, it takes $O(k_{max}L)$ time at most. The operations in steps 19–42 take O(m). Therefore, the total complexity becomes $O(mN_RN \log N + mN_RL + k_{max}N \log(k_{max}N) + k_{max}L(N + m))$.

3-1-3 Heuristic Algorithms

The exact algorithm given earlier for the dedicated impairment-aware survivable routing problem has a factorial time complexity. Thus, it is desirable to find heuristic algorithms which have better time complexity than the exact algorithm. We have provided two heuristic algorithms, D_H1ISRA and D_H2ISRA, to solve this problem, which are given below.

Algorithm 9 $D\_H1ISRA(G, m, s, d)$

```
Clone graph G (node i becomes node i + N) to graph G'
Connect the two graphs, G and G', via a link between node d and node s + N with
   weight of Δ and make graph G''.
M_HSAMCRA(G'', m, s, d') {find a path between s and d' = d + N in graph G''}
```

Algorithm D_H1ISRA is a modification of the exact algorithm, D_EISRA, in which the number of paths a node can store, k_{max}, is limited to 2. Even though limiting k_{max} attains a better time performance, the optimality of the solution will be lost. The meta code is the same as that of D_EISRA except at step 37 of M_HSAMCRA where it checks if the new extended path satisfies the look-ahead conditions, obeys the physical constraint and the counter of the node (number of paths stored at that node). Once D_H1ISRA creates the graph on which the path computation process is performed, it calls the subroutine, M_HSAMCRA (Modified Heuristic SAMCRA), which performs the operation.

The complexity of algorithm D_H1ISRA is obtained from complexity of D_EISRA by replacing k_{max} by 2, which is $O(mN_RN \log N + mN_RL + N \log N + L(N + m))$.

Algorithm 10 $M\_HSAMCRA(G, m, s, d)$

for each $v \in V$ **do**
  counter$[v] \leftarrow 0$
end for
maxlen $\leftarrow 1.0$
for $i = 1, \ldots, m$ **do**
  DIJKSTRA$(G, \{N_R, d\}, i) \rightarrow b_i(n)$
end for
queue $Z \leftarrow 0$
counter$[s] \leftarrow$ counter$[s]$ + 1
$last\_reg \leftarrow 0$
INSERT $(Z$,num,0.0,$l(\overrightarrow{b}(s)), 0, 0, last\_reg, 0, s, counter$[s])
while $(Z \neq 0)$ **do**
  EXTRACT-MIN$(Z) \rightarrow u[i]$
  $u[i] \leftarrow$ GREY
  if $(u = d)$ **then**
    STOP$\rightarrow$ return path
  else
    for each $v \in$Adj$[u]$ AND $v \notin \{\pi[u[i]], s\}$ **do**
      $flag_{ru} \leftarrow 0$
      $\overrightarrow{pc}_{slr} \leftarrow \overrightarrow{I^*}(u[i]) + \overrightarrow{q}_{u \rightarrow v}$
      $\overrightarrow{pc} \leftarrow \overrightarrow{I}(u[i]) + \overrightarrow{q}_{u \rightarrow v}$
      if $(l(\overrightarrow{pc}) > maxlen)$ **then**
        if (lur$(u[i]) \neq 0)$ **then**
          $\overrightarrow{pc} \leftarrow \overrightarrow{pc}_{slr}$
          $flag_{ru} \leftarrow 1$
          $reg(u[i] + N) \leftarrow reg(u[i] + N) - 1$
        end if
      end if
      if $(flag_{ru} = 1)$ **then**
        $reg\_count \leftarrow reg\_numb(u[i]) + 1$
        $last\_reg \leftarrow 0$
      else
        $reg\_count \leftarrow reg\_numb(u[i])$
        $last\_reg \leftarrow lur(u[i])$
      end if
      look_aheadPC$\leftarrow l(\overrightarrow{I^*}(u[i]) + \overrightarrow{q}_{u \rightarrow v} + \overrightarrow{b^*}[v])$
      if $((look\_aheadPC \leq maxlen$ AND $(l(\overrightarrow{pc}) \leq maxlen)$ AND $counter[v] \leq k_{max}$ **then**
        if $reg(v) \neq 0$ **then**
          $last\_reg \leftarrow v$
          $\overrightarrow{pc}_{slr} \leftarrow \overrightarrow{0}$
        end if
        counter$[v]\leftarrow$ counter$[v]$+1
        INSERT$(Z$,num,prediction,reg_count,$l(\overrightarrow{pc}), last\_reg, l(\overrightarrow{pc}_{slr})$,$v$,counter$[v])$
        $\overrightarrow{I}(v[counter[v]]) \leftarrow \overrightarrow{pc}$
        $lur(v[counter[v]]) \leftarrow last\_reg$
        $\overrightarrow{I^*}(v[counter[v]]) \leftarrow \overrightarrow{pc}_{slr}$
        $\pi[v[$counter$[v]]] \leftarrow u[i]$
      end if
    end for
  end if
end while

The other heuristic algorithm, D_H2ISRA, is given below. It uses principles of Suurballes's algorithm in addition to other methods that used to increase its performance. Algorithm D_H2ISRA starts the computation of feasible survivable path by connecting all nodes $u, v \in N_R \cup \{s, d\}$ that are within a distance of less than or equal to the physical impairment threshold using Dijkstra's shortest path algorithm on graph G, creating graph G' (a graph that only consists of the regenerator nodes, the source node and the destination node) during the process. This approach is chosen due to its ability to use regenerators during the path computation process. i.e., all the intermediate nodes of the shortest path that is computed on graph G' are regenerator nodes that are placed within a distance of less than or equal to the physical impairment threshold.

In step 3 the shortest path between the source and destination nodes is computed in graph G'. Since graph G' is a sub-graph of the original graph, links of the path computed in step 3 are replaced by the corresponding sub-paths in G in step 4. However, since this path is made of a concatenation of path segments, it may not be a simple path. Hence, its loops are removed in step 5. Once the primary path, $P^1_{s\to d}$, is found, the next step is checking its feasibility in step 6. Even though the path computed on graph G' is feasible, it may be infeasible once its loops are removed. If the primary path is not feasible, the loops (links that make up the loop) of path $P^1_{s\to d}$ are removed from graph G one by one and graph G''' is constructed. Steps 1–6 are repeated on graph G''' until a feasible path is found or all loops of the first path computed are removed. If a feasible primary path is found, the links of $P^1_{s\to d}$ are redirected from d to s to obtain graph G'' in step 15. The same procedures used to find $P^1_{s\to d}$ are repeated on graph G'' to find a feasible backup path that is link-disjoint with the primary path.

Complexity of D_H2ISRA

The major operations that determine the complexity of algorithm D_H2ISRA are: constructing sub-graphs G', G'', G''' and G'''', finding the shortest paths between s and d in these graphs and removing the loops of the paths in G. The construction of the graphs involves finding the shortest paths between each pair of nodes. This can be implemented with a complexity of $O(N_R^2 \log N_R + N_R L)$. Since the computation complexity of Dijkstra's algorithm is $O(N_R^2)$, finding the shortest paths in G' and G'' will have a complexity of $O(N_R^2)$. The total complexity of removing loops in step 5 is $O(N^2)$. If the computed path is not feasible, step 13 removes the loops and the previous operations will be performed again. Thus, the worst complexity of this operation is $O(l_c(N_R^2 \log N_R + N_R L + N^2))$. Therefore, the total complexity of algorithm D_H2ISRA is $O(l_c(N_R^2 \log N_R + N_R L + N^2))$.

Algorithm 11 $D\_H2ISRA(G, s, d, \Delta)$

1: Find the shortest path $\{P^*_{u\to v}\}$ between all nodes $u, v \in N_R \cup \{s, d\}$, for which $r(P^*_{u\to v}) \leq \Delta$.
2: Make a graph $G'(\mathcal{N}, \mathcal{L}')$, where $L' = \{(u, v) \mid r(P^*_{u\to v}) \leq \Delta\}$ and assign the links the cost of the actual path.
3: Find the shortest path $P'_{s\to d}$ from s to d in G'.
4: Substitute the links of $P'_{s\to d}$ with the corresponding sub-paths $P^*_{u\to v}$ in G to obtain $P_{s\to d}$.
5: Remove all the loops of $P_{s\to d}$ in graph G to obtain path $P^1_{s\to d}$.
6: Check feasibility of path $P^1_{s\to d}$.
7: **if** feasible **then**
8:     goto 15
9: **else**
10:     goto 12
11: **end if**
12: **for** $i = 1$ to $i = l_c\ of\ P_{s\to d}$ **do**
13:     Remove loop i in graph G and make graph G'''. Repeat the above steps on graph G'''.
14: **end for**
15: Redirect all links in $P^1_{s\to d}$ from d to s on the original graph to obtain $G''(\mathcal{N}, \mathcal{L}'')$.
16: On graph G'' repeat steps $1-5$ to obtain path $P^2_{s\to d}$.
17: Check feasibility of path $P^2_{s\to d}$.
18: **if** feasible **then**
19:     goto 26
20: **else**
21:     goto 23
22: **end if**
23: **for** $i = 1$ to $i = l_c\ of\ P^2_{s\to d}$ **do**
24:     Remove loop i in graph G'' and make graph G''''. Repeat the steps starting from 16 on graph G''''.
25: **end for**
26: Remove all links that are both in $P^1_{s\to d}$ and $P^2_{s\to d}$ to obtain link-disjoint paths.
27: **if** feasible primary and backup paths exist **then**
28:     accept the request
29: **else**
30:     reject the request
31: **end if**

3-1-4 Simulation Results

This section discusses the simulation results of the algorithms discussed earlier in this chapter. In our simulations, our objective is to find primary and backup paths for a single connection request in translucent WDM network. A connection request is represented by a request-id, a source, a destination, and physical impairment constraints. Each link in a network is associated with one physical impairment which is assigned a random value between 0 and 1 by a uniformly distributed random function. The regenerators are also randomly distributed in the network. In this problem, it is assumed that there are enough wavelengths so that the wavelength continuity constraint is relaxed.

The goal of the simulation is to get a better insight into the performance of the algorithms and make a comparison between the algorithms while varying the physical impairment threshold and the number of regenerator nodes in the network. The performance metrics used to compare the algorithms are: running time, average regenerator number of the path and the acceptance ratio. Running time is the time taken by the algorithms to process the connection request, the average regenerator number is the average number of regenerators used to set up the connection and acceptance ratio is the ratio of the number of accepted requests to the total number of requests.

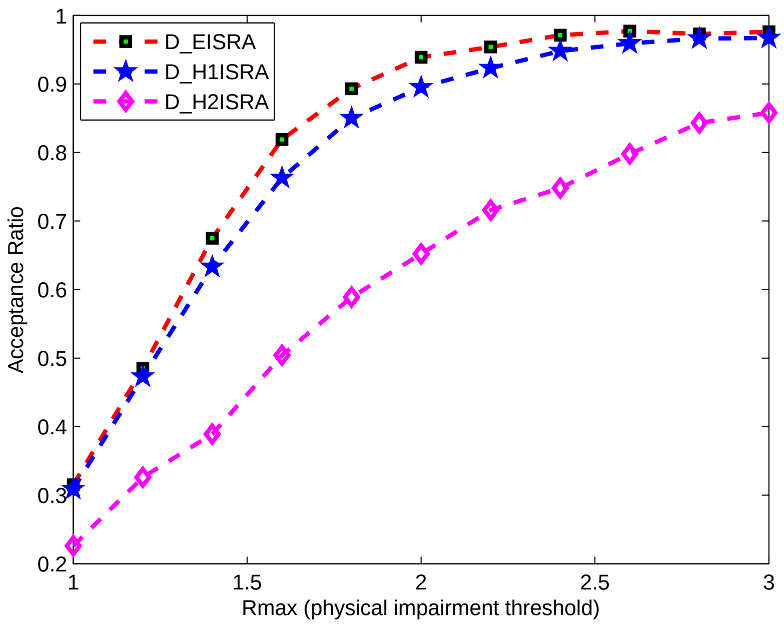

Figure 3-2: Acceptance ratio vs physical impairment threshold for the ARPANET network with $N_R = 12$.

Figure 3-2 shows the acceptance ratio vs physical impairment threshold for the ARPANET network with $N_R = 12$. For this network, as the physical impairment threshold increases the acceptance ratio also increases. This is because, as the physical impairment threshold increases less number of paths will fail to satisfy the physical impairment constraint, which results in acceptance of more requests. The figure depicts that D_H2ISRA obtains a lower acceptance ratio compared to D_EISRA and D_H1ISRA. Also, the exact algorithm (D_EISRA) has slightly better acceptance ratio than its heuristic counterpart (D_H1ISRA). The difference between the algorithms in this performance metric stems from the fact that D_EISRA exhaustively explores all the possibilities in order to obtain the shortest feasible path; whereas, D_H1ISRA explores few of the possibilities since it stores only two paths at a node while D_H2ISRA only considers few possibilities, i.e., once it gets a feasible primary path it will start computing the backup path which may lead to unsuccessful computation even though there could be another primary path which could lead to finding a feasible backup path.

Additional simulations are also performed on NSFNET (Figure A-9) and lattice (Figure A-10) networks of $N = 49$. The results of these simulations also show similar patterns as the ARPANET network.

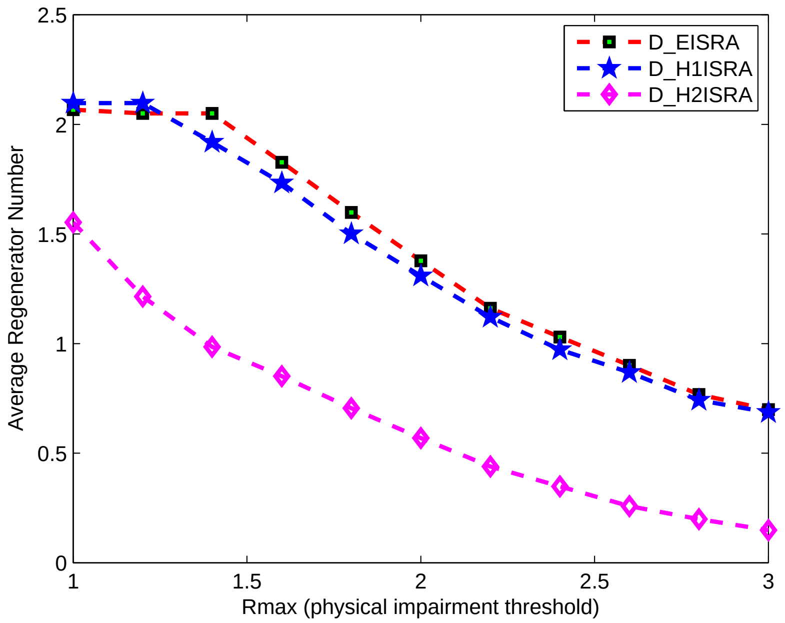

Figure 3-3: Average regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 12$.

Figure 3-3 shows the average regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 12$. The figure depicts that when the physical impairment threshold increases the average regenerator number decreases. When the physical impairment is small, a path traverses few nodes before it violates the physical impairment constraint. Which leads to the necessity of signal regeneration at small intervals. Since the number of regenerators in the network is fixed, only shorter paths that require few regenerators are accepted resulting in small average path regenerator number. Whereas, when the physical impairment is large requests will be accepted without the need of frequent signal regeneration which results in a decrease of the average regenerator number required to accept the request. It can be observed that D_H2ISRA attains a smaller average regenerator number compared to D_EISRA and D_H1ISRA since its acceptance ratio is less than the other two algorithms. In addition, D_H1ISRA may some times attains a larger average regenerator number compared to D_EISRA. This result may have arisen from the fact that D_H1ISRA may have stored a sub-path that leads to a path which uses large number of regenerators since it can only store limited paths at a node.

The simulations conducted on NSFNET and lattice networks with these algorithms also show similar behaviors as the ARPANET network. The results of these simulations are given in Figure A-11 and Figure A-12.

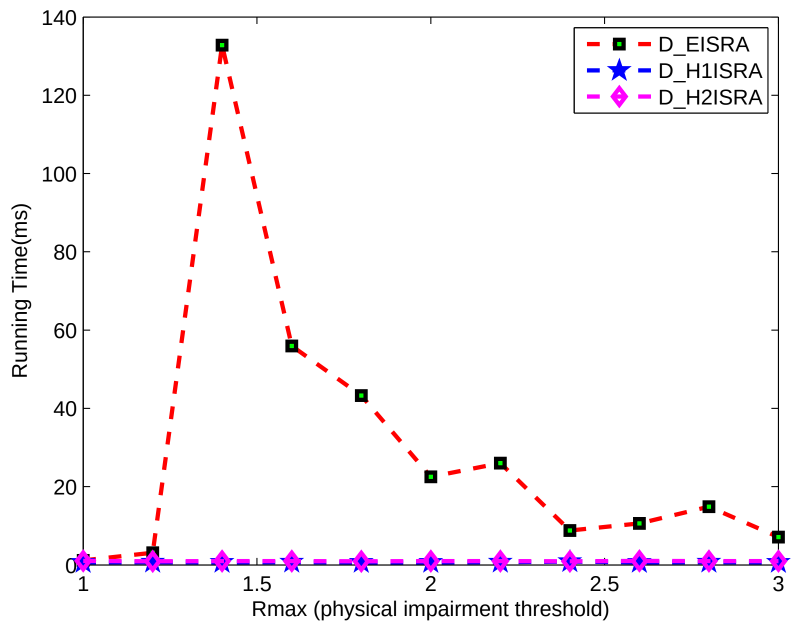

Figure 3-4: Running time vs physical impairment threshold for the ARPANET network with $N_R = 12$.

The running time vs physical impairment threshold for the ARPANET network with $N_R = 12$ is shown in Figure 3-4. The figure shows that D_EISRA has a larger running time compared to D_H1ISRA and D_H2ISRA since it runs extensive search procedure to find a feasible path. As shown in the figure, the running time of D_EISRA becomes large in the region where the physical impairment threshold is medium.

When the physical impairment threshold is small, since a path traverses only few nodes before it violates the physical impairment threshold, the path requires regeneration at small intervals. But the probability of finding a regenerator in a small interval is less due to scarcity of regenerators, which results in less probability of regeneration. As a result, more paths will be dropped thereby reducing the running time.

When the physical impairment threshold is large, paths are found easily which results in the decrease of running time. However, when the physical impairment threshold is medium, the probability that a path finds a regenerator node before it violates the physical impairment threshold is higher compared to when physical impairment threshold is small. Which reduces the number of paths dropped. Furthermore, paths are not found as easily as when the physical impairment threshold is large. Which leads to a more exhaustive search that stores large number of paths at nodes, resulting in long running time. The running time becomes even longer when the path can not be found.

The simulations conducted on lattice network with these algorithms also show similar behaviors as the ARPANET network. Whereas for NSFNET network, unlike the ARPANET network the running time increases with the physical impairment threshold where the spike of D_EISRA in the medium constrained region does not exist. This difference arises due to the fact that NSFNET network is a network with smaller number of nodes and links, resulting in a smaller search space. The additional results of these simulations are given in Figure A-13 and Figure A-14 for NSFNET and lattice networks.

Figure 3-5 shows the acceptance ratio vs total number of regenerators for the ARPANET network. As shown in the plot, as the total number of regenerators increases more requests are accepted, since there will be more probability of finding a regenerator whenever regeneration is required. The figure reveals that D_EISRA and D_H1ISRA outperform D_H2ISRA in terms of acceptance ratio due to the same reason discussed earlier. The figure also depicts that D_EISRA provides slightly better acceptance ratio over D_H1ISRA. Plots shown in Figure A-15 and Figure A-16 for NSFNET and lattice networks respectively also show the same behavior as the ARPANET network.

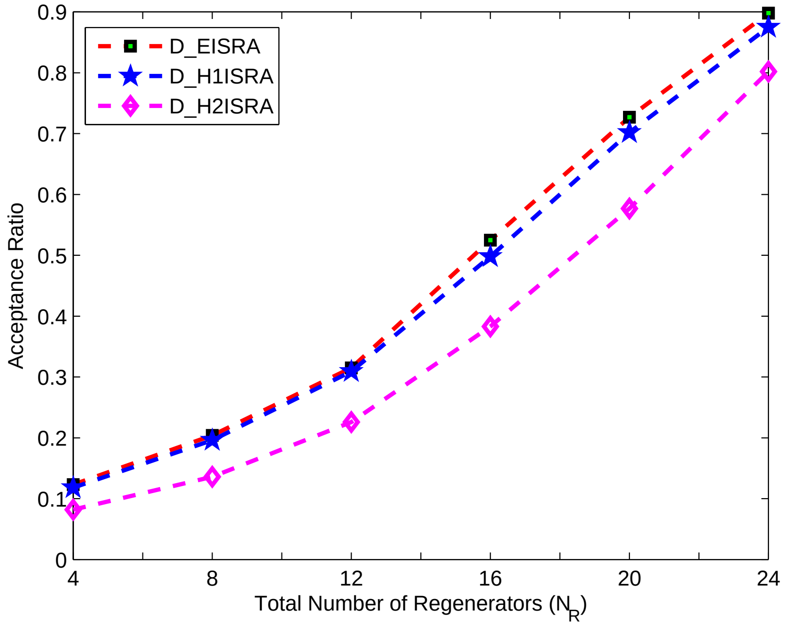

Figure 3-5: Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

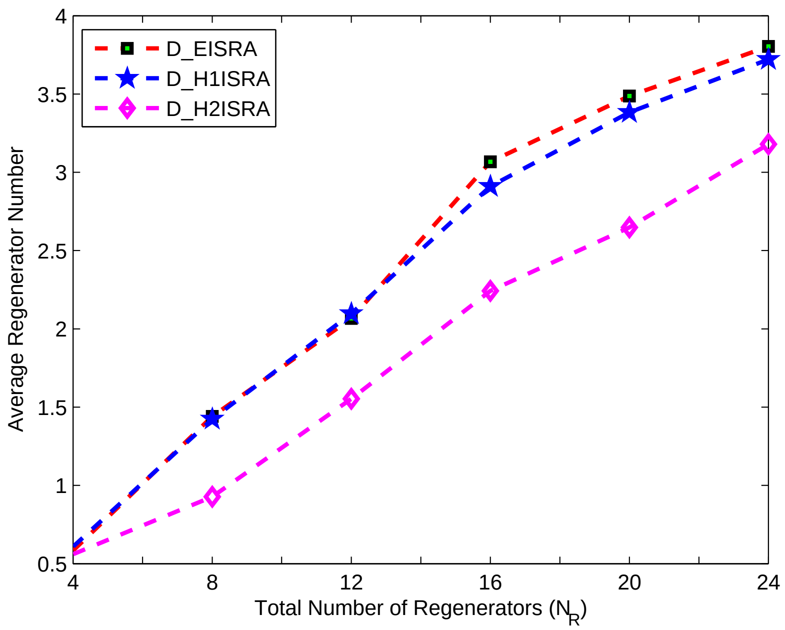

Figure 3-6: Average regenerator number vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

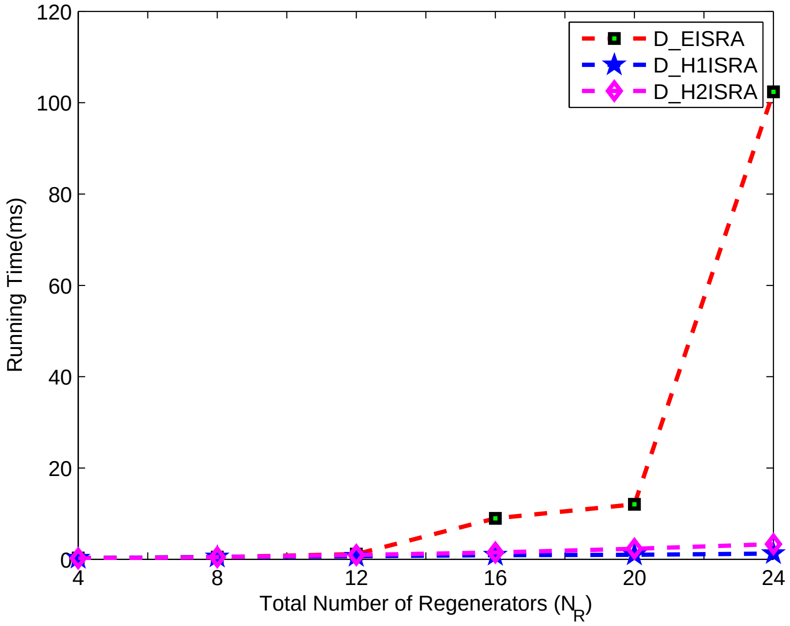

Figure 3-7: Running time vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

The plot of the average regenerator number vs total number of regenerators in the network shown in Figure 3-6 is obtained for the ARPANET network. As shown in the plot, the average regenerator number increases with the total number of regenerators in the network. This arises from the fact that increasing the number of regenerator nodes in a network for small physical impairment threshold leads to the acceptance of paths that require more regenerators, increasing the average regenerator number. i.e., for small physical impairment threshold when there are only few regenerators, most paths will be dropped due to lack of regeneration, whereas, when the number of regenerators increases those paths will be accepted increasing the average regenerator number in the process. Additional results shown in Figure A-17 and Figure A-18 also show the same behavior as the ARPANET network.

As can be observed from Figure 3-7, the running time of the algorithms increases with the total number of regenerators in the network. The figure shows that D_EISRA has a larger running time compared to D_H1ISRA and D_H2ISRA due to the same reason discussed earlier. As shown in the figure, for the ARPANET network the running time of D_EISRA becomes large in the region where the total number of regenerators in the network is large. When the total number of regenerators in the network is small it is difficult to find regenerators when regeneration is required, which causes more paths to be dropped thereby reducing the running time. When the total number of regenerators in the network increases a path can find a regenerator node before it violates the physical impairment threshold. As a result, the number of paths dropped will be reduced. Which leads to a more exhaustive search that stores larger number of paths at nodes, resulting in long running time. In addition, the time spent computing the physical impairment lower bound vector also increases with the number of regenerator nodes in the network for D_EISRA and D_H1ISRA algorithms. In case of D_H2ISRA, as the number of regenerator nodes in the network increases the time spent building the graph containing the source, the destination and the regenerator nodes also increases. Which leads to the overall increase in the running time. Figure A-19 and Figure A-20 also show the same behavior for NSFNET and lattice networks respectively.

3-2 Shared Impairment-aware Survivable Routing Problem

In this sub-problem, regenerators are shared between the primary and backup lightpaths of a request. As discussed earlier, this type of shared regenerator assignment is used in a scenario in which the network is protected by the 1:1 protection scheme where both the primary and backup lightpaths do not need regenerators at the same time. i.e., only one of the paths are carrying a signal at a time. Since we are considering shared protection between primary and backup of the same request, the different requests do not share regenerators. Thus, each request can be considered individually and the problem can be reduced to a single request shared impairment-aware survivable routing problem.

3-2-1 Problem Definition

1. **Given:** A network topology which is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of N nodes and $\mathcal{L}$ is the set of L links between the nodes; associated with each link $(u, v) \in \mathcal{L}$ there is a wavelength and a set of non-negative physical impairment metrics $r_i(u, v)$ for $i = 1, \ldots, m$. $\mathcal{N}_R \subseteq \mathcal{N}$ represents the set of R nodes that have regeneration capability, and $N_R = |\mathcal{N}_R|$ represents the number of regenerator nodes. A lightpath request is represented by the tuple (s, d, Δ), where $s, d \in \mathcal{N}$ are the source and destination nodes of the request and Δ is the set of threshold values of the physical impairments.

2. **Objective:** Find feasible link-disjoint primary and backup simple paths that share regenerators from the source node to the destination node, i.e., the two paths share regenerators and do not violate the physical impairment constraint.

To solve the shared impairment-aware survivable routing problem, we modified the three algorithms, D_EISRA, D_H1ISRA and D_H2ISRA, given for the dedicated impairment-aware survivable routing problem earlier in this chapter into algorithms S_EISRA, S_H1ISRA and S_H2ISRA respectively. Since the meta codes of the algorithms S_EISRA, S_H1ISRA and S_H2ISRA are the same as their dedicated counterparts D_EISRA, D_H1ISRA and D_H2ISRA respectively, except for minor changes during implementation, we will not discuss them here. The minor modification done is that, while computing the backup path, if the computation process finds a regenerator used by the primary path it will reuse it unlike the dedicated backup computation process. Therefore, the complexity of these algorithms is the same as the ones given earlier for the dedicated impairment-aware survivable routing algorithms. The simulation results are given in the following sub-section.

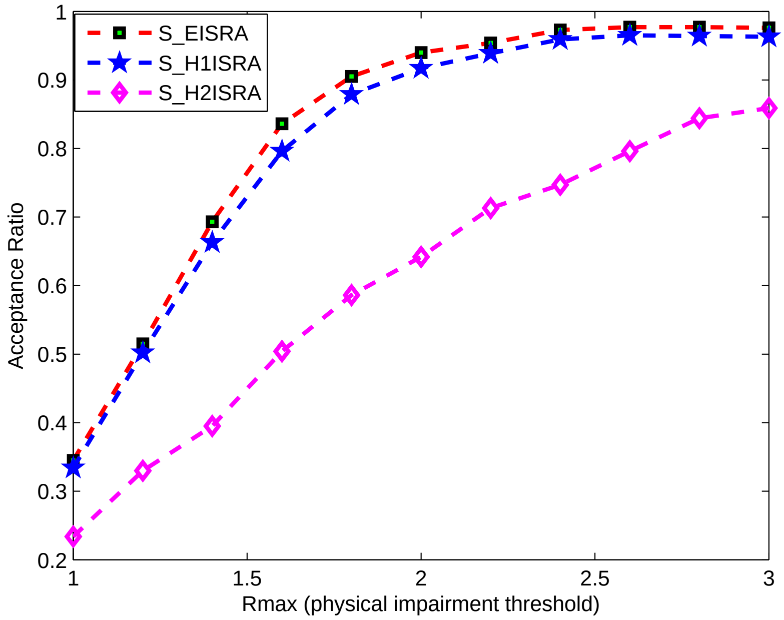

Figure 3-8: Acceptance Ratio vs physical impairment threshold for the ARPANET network with $N_R = 12$ (shared regenerator assignment between primary and backup path).

3-2-2 Simulation Results

This section discusses the simulation results of the algorithms discussed earlier to solve the shared impairment-aware survivable routing problem. In our simulations, our objective is to find primary and backup paths which share regenerators for a single connection request in translucent WDM network. A connection request is represented as described in the previous section. A network's link is associated with one physical impairment which is assigned a random value between 0 and 1 by a uniformly distributed random function. The regenerators are also randomly distributed in the network. Just like earlier simulations, it is assumed that there are enough wavelengths so that the wavelength continuity constraint is relaxed.

The simulations are performed to see the performance of the algorithms and make a comparison between the algorithms while varying the physical impairment threshold and the number of regenerator nodes. The algorithms are compared based on the running time, average regenerator number of the path and the acceptance ratio.

Figure 3-8 shows the acceptance ratio vs physical impairment threshold for the ARPANET network. The algorithms compared in the figure are the shared versions of previously described dedicated algorithms. As shown in the figure, the properties of the algorithms are the same as described earlier for the dedicated ones. The figure depicts that S_H2ISRA

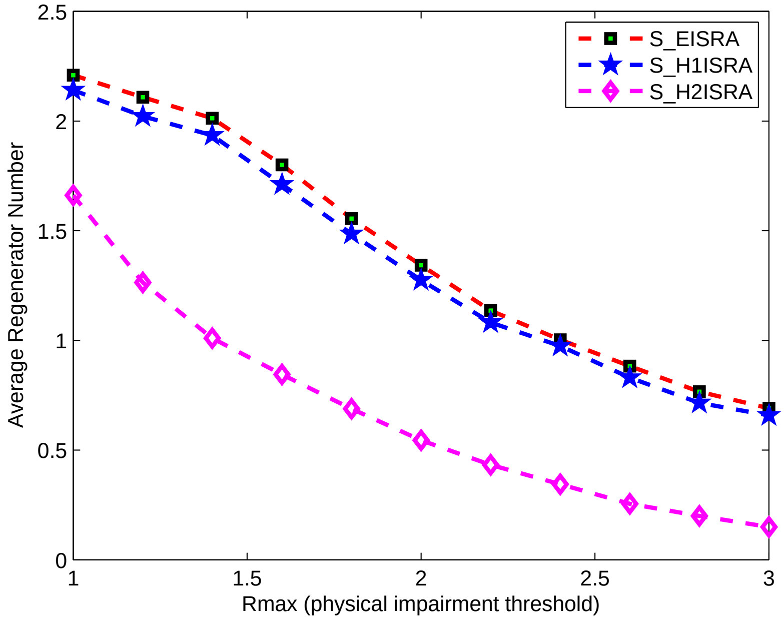

Figure 3-9: Average regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 12$ (shared regenerator assignment between primary and backup path).

obtains a lower acceptance ratio compared to S_EISRA and S_H1ISRA for the same reason discussed earlier. Also, the exact algorithm (S_EISRA) has slightly better acceptance ratio than its heuristic counterpart (S_H1ISRA).

Figure 3-9 shows the average regenerator number vs physical impairment threshold for the ARPANET network. It can be observed that S_H2ISRA attains a smaller average regenerator number compared to S_EISRA and S_H2ISRA. As shown in the figure, the algorithms behave as described earlier for the dedicated algorithms when the physical impairment threshold is varying.

The running time vs physical impairment threshold of the algorithms is given in Figure 3-10 for the ARPANET network with $N_R = 12$. The plot has the same characteristics as described above for the dedicated algorithms. The figure shows that S_EISRA has a larger running time compared to S_H1ISRA and S_H2ISRA for the same reasons discussed in earlier sections.

Figure 3-11 shows the acceptance ratio vs total number of regenerators for the ARPANET network. As shown in the plot, as the total number of regenerators increases more requests are accepted. This result arises due to the fact that whenever regeneration is required acquiring a regenerator becomes easier as the number of regenerators increases. Which

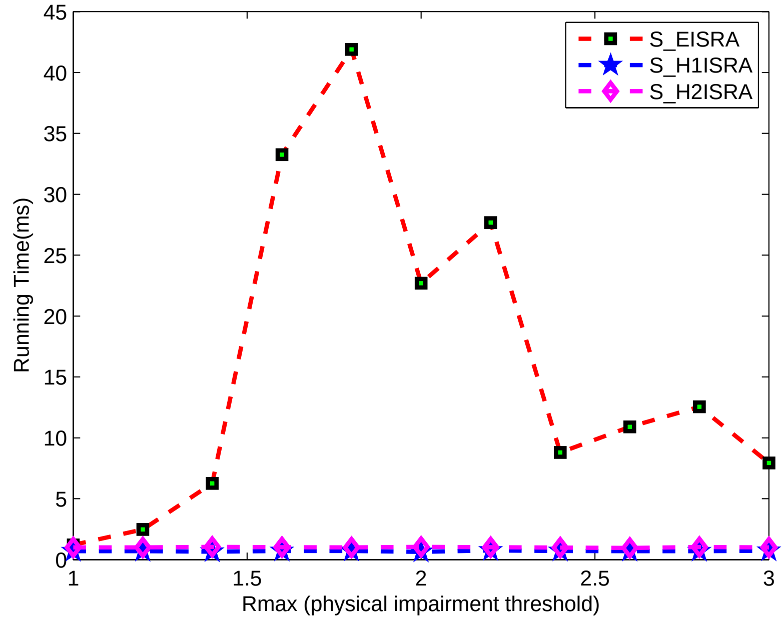

Figure 3-10: Running time vs physical impairment threshold for the ARPANET network with $N_R = 12$ (shared regenerator assignment between primary and backup path).

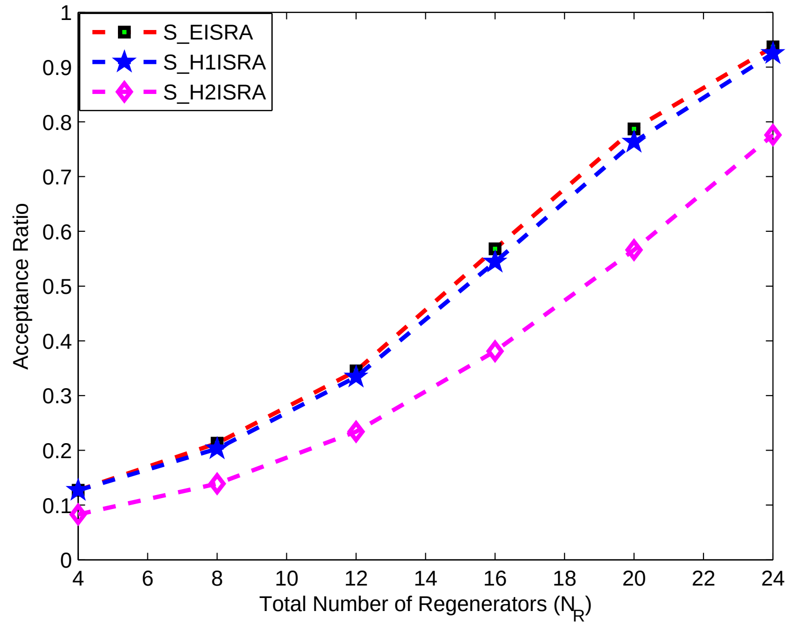

Figure 3-11: Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold $= 1$ (shared regenerator assignment between primary and backup path).

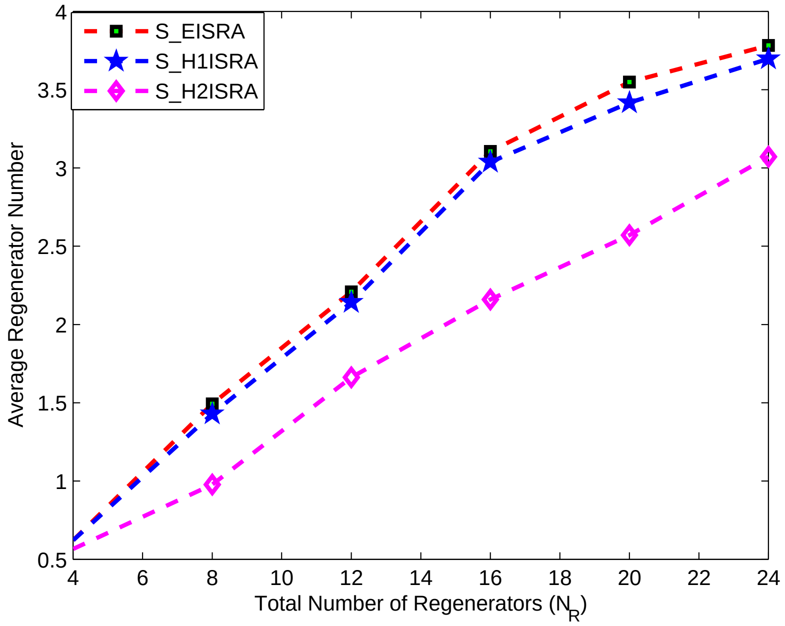

Figure 3-12: Average regenerator number vs total number of regenerators for the ARPANET network with physical impairment threshold = 1 (shared regenerator assignment between primary and backup path).

leads to the reduction of paths that will be dropped. The figure reveals that S_EISRA and S_H1ISRA outperform S_H2ISRA in terms of acceptance ratio due to the same reasoning discussed in previous sections. The figure also depicts that S_EISRA provides slightly better acceptance ratio compared to S_H1ISRA.

As can be observed from Figure 3-12, the average regenerator number increases with the total number of regenerators in the network. This arises from the fact that increasing the number of regenerators in a network for small physical impairment threshold leads to the acceptance of paths that require more regenerators, increasing the average regenerator number. The figure also shows that S_H2ISRA has smaller average number of regenerators.

As can be observed from Figure 3-13 the running time of the algorithms increases with the total number of regenerators in the network. The figure shows that S_EISRA has a larger running time compared to S_H1ISRA and S_H2ISRA. As shown in the figure, the running time of S_EISRA becomes large in the region where the total number of regenerators in the network is large. When the total number of regenerators in the network is small, it is difficult to find regenerators when the physical impairment of the path exceeds the physical impairment threshold which causes more paths to be dropped and fewer paths to be stored thereby reducing the running time. When the total number of regenerators in the network increases, a path can find a regenerator node before it violates the physical impairment threshold. As a result, the number of paths dropped will be reduced. Which leads to

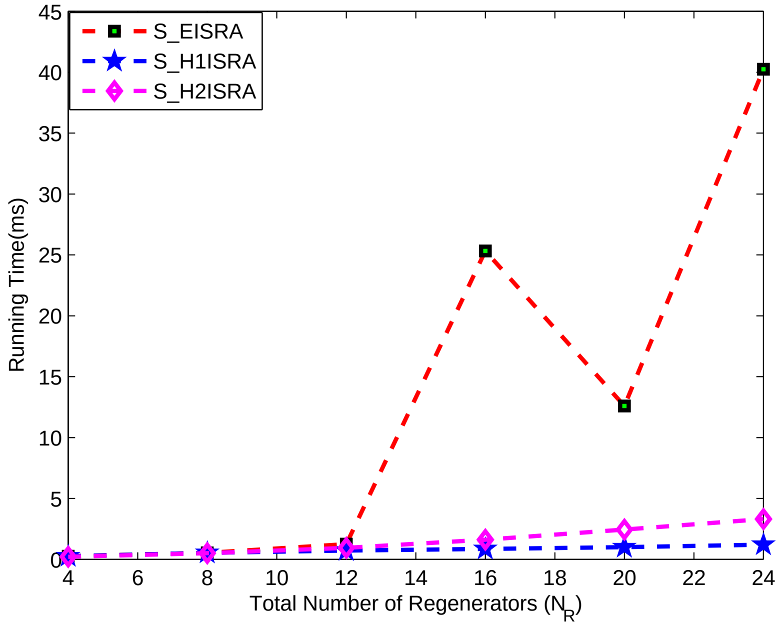

Figure 3-13: Running time vs total number of regenerators for the ARPANET network with physical impairment threshold = 1 (shared regenerator assignment between primary and backup path).

a more exhaustive search that stores larger number of paths at nodes, resulting in long running time.

3-3 Backup Shared Impairment-aware Survivable Routing Problem

In this sub-problem, regenerators are shared between backup lightpaths as long as their primary paths do not fail simultaneously. Since only a single link failure is considered in our study, as far as primary paths are link-disjoint they will not fail at the same time. For this problem, we provide one exact algorithm, BS_EISRA, and two heuristic algorithms, BS_H1ISRA and BS_H2ISRA.

3-3-1 Problem Definition

1. **Given:** A network topology which is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of N nodes and $\mathcal{L}$ is the set of L links between the nodes; associated with each link $(u, v) \in \mathcal{L}$ there is a wavelength and a set of non-negative physical impairment metrics $r_i(u, v)$ for $i = 1, \ldots, m$. $\mathcal{N}_R \subseteq \mathcal{N}$ represents the set of R nodes that have regeneration capability, and $N_R = |\mathcal{N}_R|$ represents the number of regenerator nodes. A lightpath request j is represented by the tuple (s_j, d_j, Δ), where $s_j, d_j \in \mathcal{N}$ are the source and destination nodes of request j and Δ is the set of threshold values of the physical impairments.

2. **Objective:** Find feasible link-disjoint primary and backup simple paths from the source node to the destination node where backup paths share regenerators, i.e., the backup paths whose primary paths are link-disjoint may share regenerators.

To solve the backup shared impairment-aware survivable routing problem, we modified the three algorithms, D_EISRA, D_H1ISRA and D_H2ISRA, devised for the dedicated impairment-aware survivable routing problem earlier in this chapter into BS_EISRA, BS_H1ISRA and BS_H2ISRA respectively. Since the meta codes of the algorithms BS_EISRA, BS_H1ISRA and BS_H2ISRA are the same as their dedicated counterparts, except for few modifications to handle regenerator sharing between backup lightpaths, we will not discuss these algorithms here. The three algorithms store a pool of free regenerators as well as regenerators used by backup paths whose primary paths are link-disjoint with primary of the request under computation. Therefore, the additional computation in these algorithms is checking disjointedness between primary paths to make sharing of regenerators possible between backup paths. During the computation process of the backup path, the algorithms check presence of regenerators at a node in the pool when regeneration is required. In addition, algorithm BS_H2ISRA sets cost of links of the backup paths whose primary paths are link-disjoint with primary of the request under computation to zero to encourage regenerator sharing. The simulation results are given in the following sub-section.

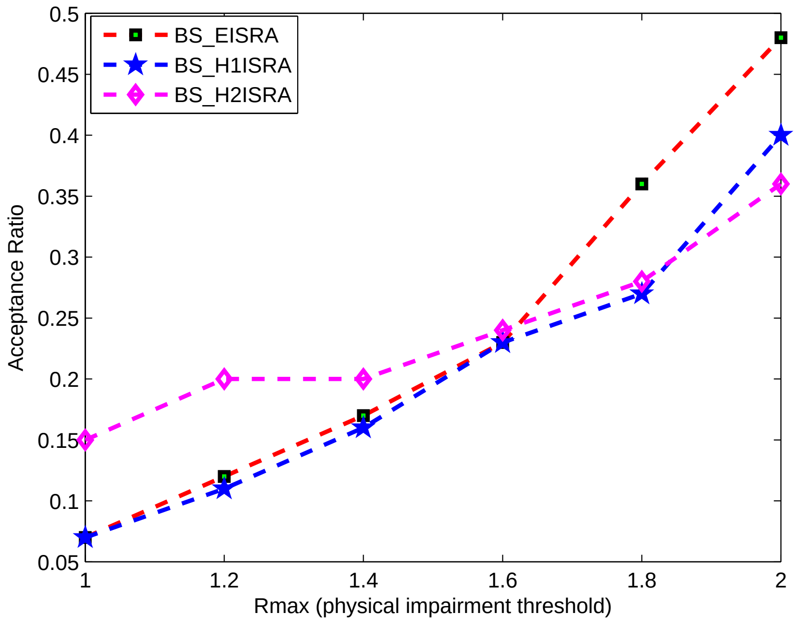

Figure 3-14: Acceptance Ratio vs physical impairment threshold for the ARPANET network with $N_R = 20$.

The exactness of the algorithm in this chapter refers to finding a feasible path if there exists. Therefore, adding additional backup path computing ability to one of the heuristic algorithms that can not be applied to the exact algorithm may make the heuristic algorithm to perform slightly better than the exact algorithm for some performance metrics.

3-3-2 Simulation Results

This section discusses the simulation results of the algorithms devised to solve the backup shared impairment-aware survivable routing problem. In our simulations, our objective is to find primary and backup paths for a connection request in translucent WDM network in which regenerator nodes can be shared between backup paths. A connection request is represented as described in the previous section. In addition, the simulation environment is the same as earlier simulation environments.

Figures 3-14 to 3-19 show simulation results of the dedicated regenerator usage algorithms modified in which the backup paths whose primary paths are link-disjoint share regenerators.

Figure 3-14 shows the acceptance ratio vs physical impairment threshold for the

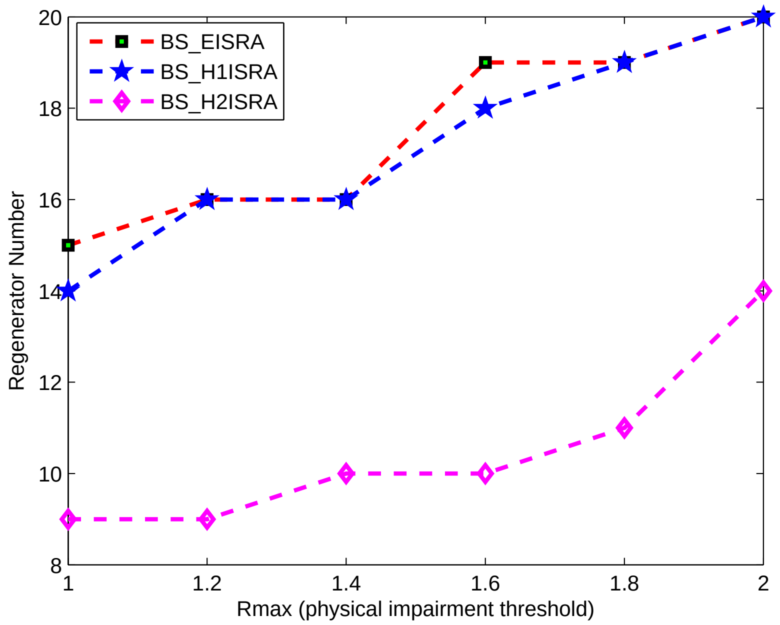

Figure 3-15: Path total regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 20$.

ARPANET network. As shown in the figure, the acceptance ratio increases with the physical impairment threshold. The reason behind this characteristic of the algorithms arises from the fact that as the physical impairment threshold increases more number of paths will satisfy the physical impairment constraint which leads to the acceptance of more number of requests.

As expected, BS_EISRA has a better performance over BS_H1ISRA in terms of acceptance ratio. Whereas, algorithm BS_H2ISRA has a greater acceptance ratio compared to the other two algorithms for small values of physical impairment threshold. This result may have been arisen from the fact that algorithm BS_H2ISRA encourages regenerator usage between backup paths whose primary paths are link-disjoint by temporarily making the weight of the links incident to the regenerator nodes of previously computed backup paths to zero unlike the other two algorithms which reuse backup path regenerators if the backup path computation process come across a regenerator used by a backup path of another flow whose primary path is link-disjoint with primary of current flow.

Figure 3-15 shows the number of regenerators used by the accepted requests vs physical impairment threshold for the ARPANET network. As shown in the plot, as the physical impairment increases more requests will be accepted since regeneration is not required in small intervals, which leads to the increase of total regenerators used. As can be seen from the graph regenerator number used by BS_H2ISRA is smaller than that of the other two

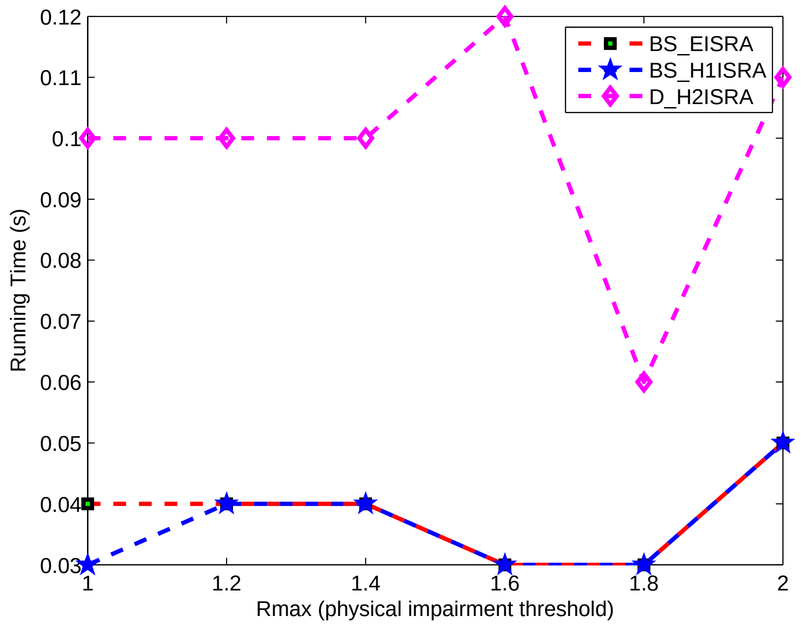

Figure 3-16: Running time vs physical impairment threshold for the ARPANET network with $N_R = 20$.

algorithms even though its acceptance ratio is larger. This shows that the acceptance ratio of the algorithm was larger for smaller physical impairment threshold values due to the acceptance of requests whose backup paths can share regenerators. As expected BS_EISRA has a bit larger regenerator number than BS_H1ISRA.

As shown in Figure 3-17, the acceptance ratio of the algorithms increases with the increase in number of regenerator nodes in the network, because less number of requests are blocked when the number of regenerators is large due to the increased probability of regeneration. Due to the same reason discussed earlier, BS_H2ISRA has a better acceptance ratio compared to BS_EISRA and BS_H1ISRA.

As can be observed from Figure 3-18, BS_H2ISRA attains the best regenerator number, whereas BS_EISRA and BS_H1ISRA attains the worst result. Since the physical impairment threshold is small, increasing the number of regenerators in a network leads to the acceptance of paths that require more number of regenerations. This leads to the increase of the average path regenerator number with the increase of the number of regenerators in a network.

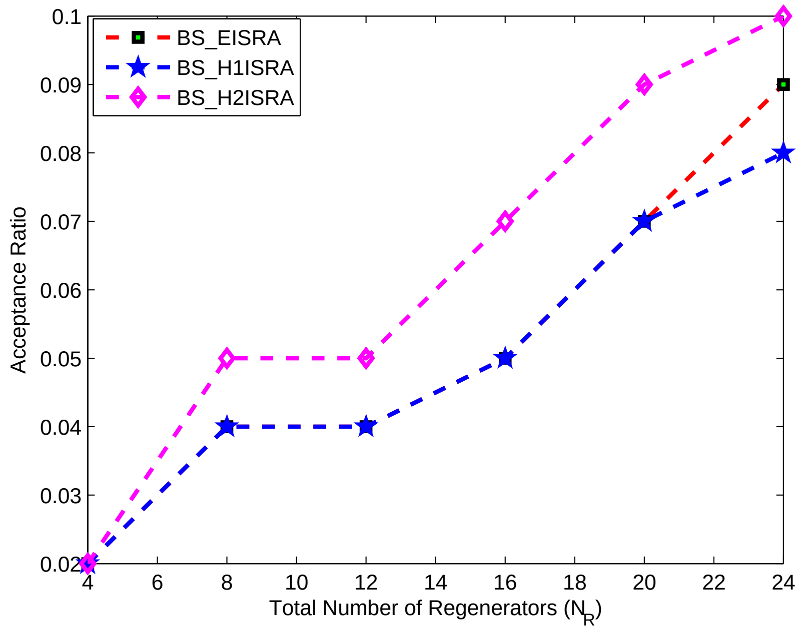

Figure 3-17: Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

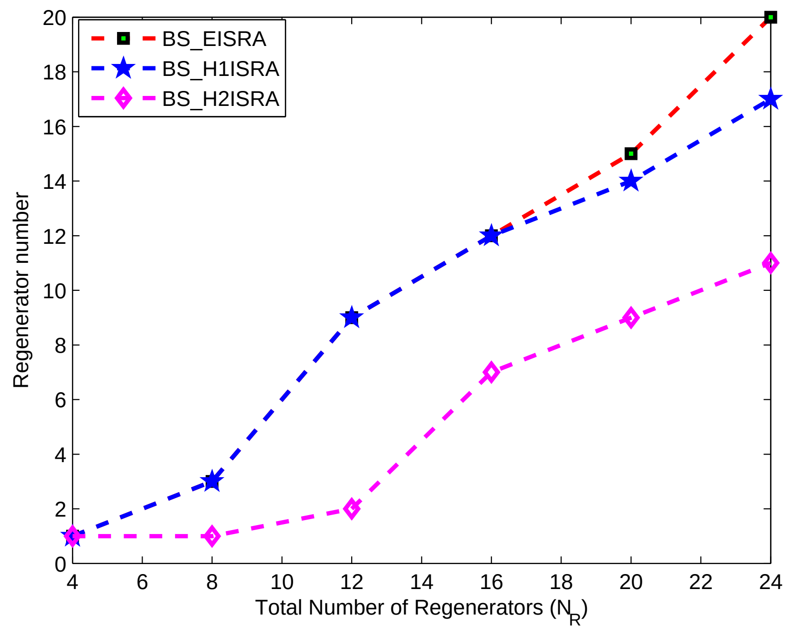

Figure 3-18: Path total regenerator number vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

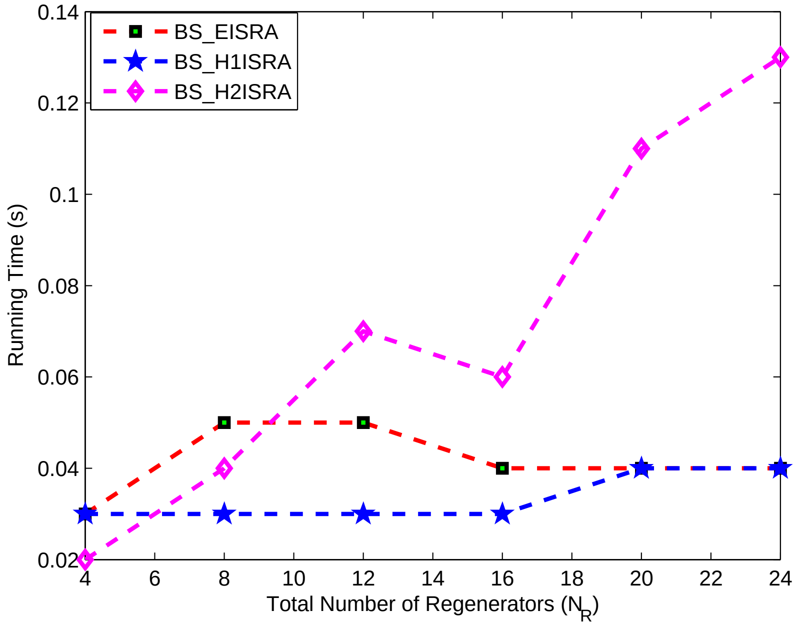

Figure 3-19: Running time vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

3-4 Impairment-aware Survivable Routing Problem under dynamic traffic

In the simulations we have been showing so far only a single connection request is considered per network. In this section the performance of the algorithms provided for the dedicated impairment-aware survivable routing problem will be analyzed under dynamic traffic. In this scenario, a dynamic traffic of connection requests is generated. Each connection request is specified by a request-id, a source, a destination, physical constraints, arrival time and departure time. The arrival and departure times of the flows are randomly generated in which the flows will be served according to their time of arrival.

Whenever a flow arrives, the routing algorithm searches for a path from the source node to the destination node based on the requirements of the flow. If the algorithm fails to find a path that satisfies the physical impairment constraint, the flow is rejected. But if a path that satisfies the constraint is found, the flow is accepted and the path is returned.

The resources used along this path, wavelength and regenerators, are reserved. Those reserved resources cannot be used by another forthcoming flow until they are made available upon the departure of the flow which reserves them.

In the simulations, flows are processed according to their arrival times. The simulations are performed on the ARPANET network with the exact and the heuristic algorithms. The performance metrics are measured by varying the value of regenerator number in the network as well as the physical impairment threshold.

For this scenario, two options are considered; one in which wavelength assignment is not an issue and another in which there are only limited wavelengths per link.

3-4-1 Unlimited wavelength

In this scenario of the problem, wavelength assignment is relaxed, i.e., it is assumed that there are enough wavelengths per link so that requests do not need to search for available wavelengths or reserve a wavelength while traversing a link.

Figures 3-20 – 3-21 show the simulation of the algorithms under dynamic traffic where wavelength availability per link is relaxed.

Figure 3-20 shows the acceptance ratio vs physical impairment threshold for the ARPANET network under dynamic traffic. As the physical impairment threshold increases the acceptance ratio also increases. This is because, as the physical impairment threshold increases less number of paths will fail to satisfy the physical impairment constraint which results

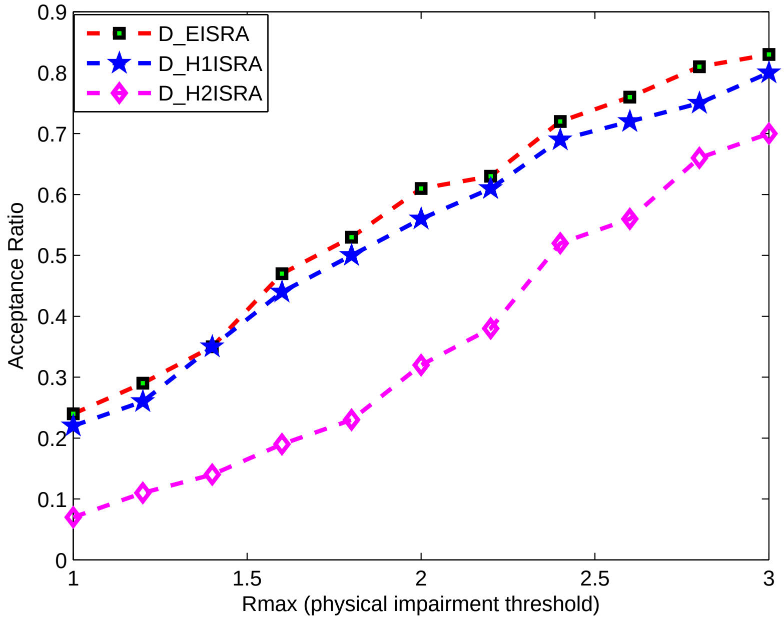

Figure 3-20: Acceptance Ratio vs physical impairment threshold for the ARPANET network with $N_R = 20$.

in acceptance of more requests. The figures depict that D_H2ISRA obtains a lower acceptance ratio compared to D_EISRA and D_H1ISRA. Also, D_EISRA has slightly better acceptance ratio over D_H1ISRA.

Figure 3-21 shows the acceptance ratio vs total number of regenerators for the ARPANET network under dynamic traffic. As shown in the plot, as the total number of regenerators increases more requests are accepted, since there will be more probability of acquiring a regenerator whenever regeneration is required. The figures reveal that D_EISRA and D_H1ISRA outperform D_H2ISRA in terms of acceptance ratio. The figures also depict that D_EISRA provides slightly better acceptance ratio over D_H1ISRA.

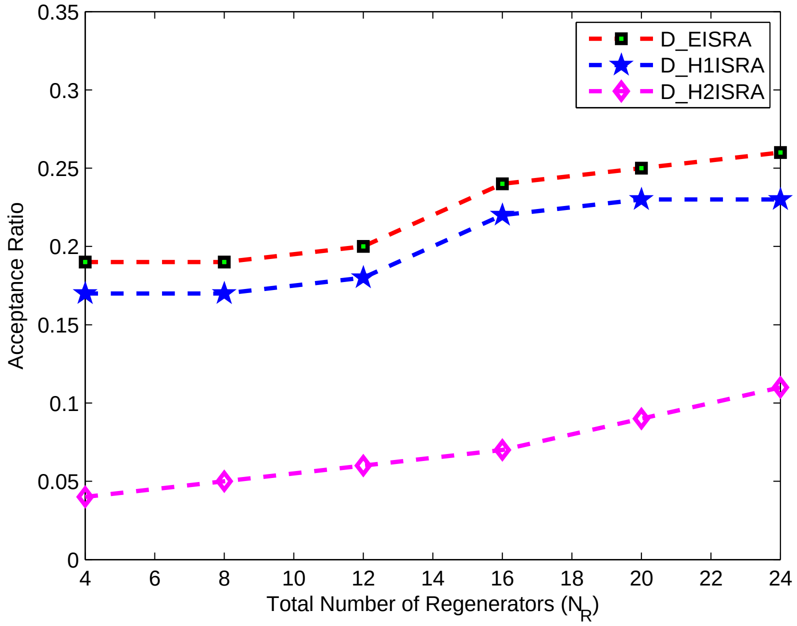

Figure 3-21: Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

3-4-2 Limited wavelength per link

In this scenario, each fiber link in a network is associated with limited wavelengths, i.e, wavelength assignment and reservation operations will be performed by the algorithms while computing a path. When a request is accepted, the wavelength used in the path is reserved at every link along this path. The wavelength used along a path is selected by the first fit (FF) approach. We used the first fit approach of selecting a wavelength due to its simplicity to implement and its practicality. First fit wavelength assignment approach does not require global knowledge about the network. Therefore, no storage is needed to keep the network states and no communication overhead is needed. These properties make its computational overhead small and its complexity low. Therefore, first fit is preferred in practice [24].

Figures 3-22 and 3-23 show the simulations of the algorithms under dynamic traffic where there are limited wavelengths per link. The figures show the same characteristics as the algorithms described earlier.

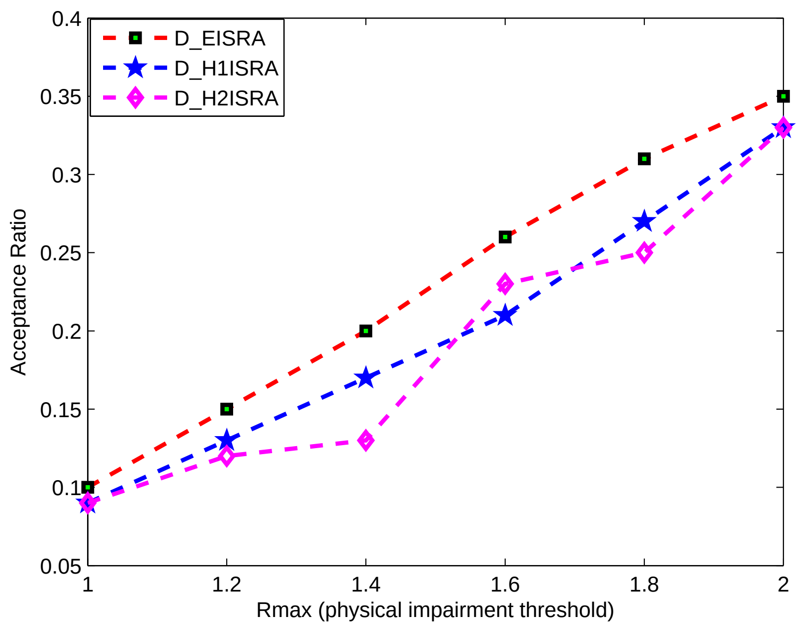

Figure 3-22: Acceptance Ratio vs physical impairment threshold for the ARPANET network with $N_R = 20$.

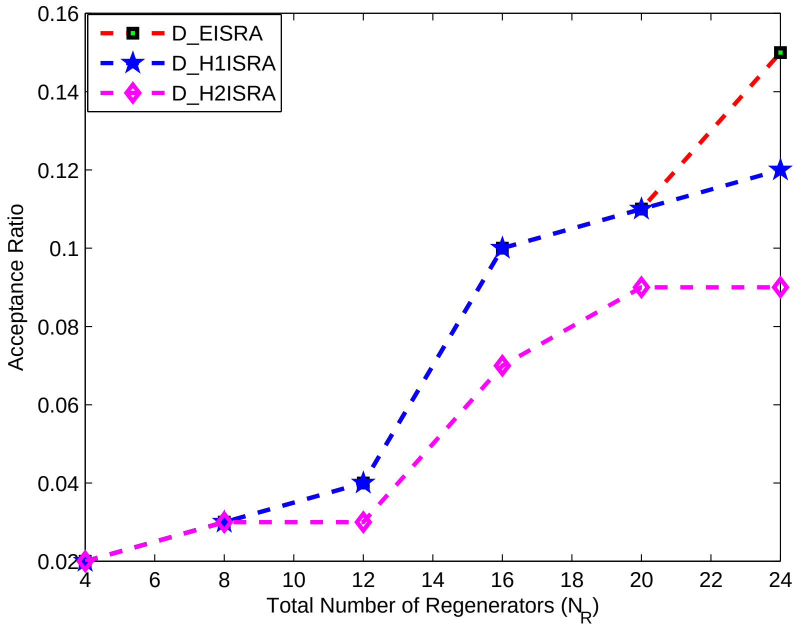

Figure 3-23: Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

3-5 Backup WR Shared Impairment-aware Survivable Routing Problem

The backup WR (wavelength regenerator) shared impairment-aware survivable routing problem is a replica of the backup shared impairment-aware survivable routing (BSISR) problem discussed earlier in this chapter. The only difference is that, here there are only limited wavelengths per link in which backup paths can share in addition to regenerators as far as their primary paths are link-disjoint. Therefore, there is no need to repeat the meta code of the algorithms in this section. We will only show the simulation results.

3-5-1 Simulation Results

In this sub-section we will show the results of the backup shared algorithms modified to accommodate regenerator as well as wavelength sharing between backup paths whose primary paths are link-disjoint. In this scenario each fiber link in a network is associated with limited wavelengths. The wavelength assignment is performed using the first fit approach due to the same reasons discussed earlier.

Figures 3-24 – 3-27 show simulation results of the modified algorithms. The figures show the usual characteristics of the algorithms as discussed in previous sections.

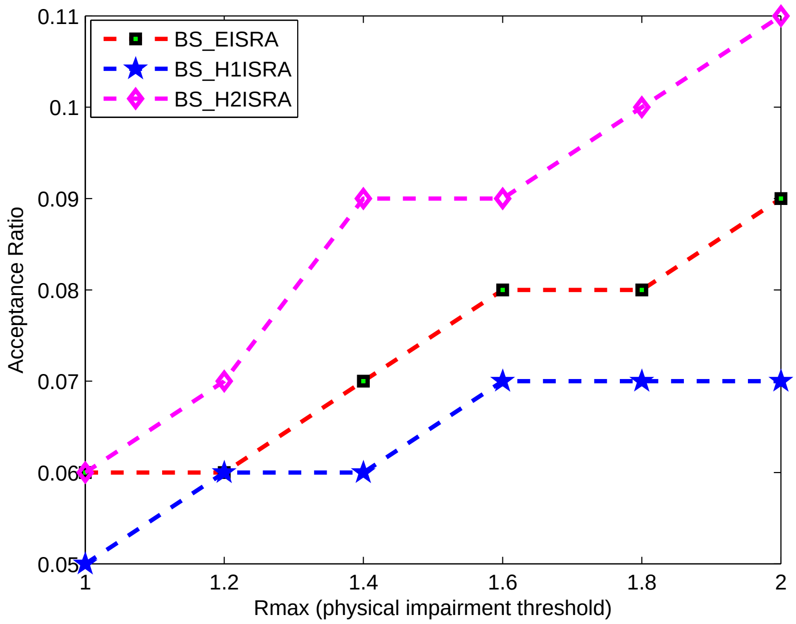

Figure 3-24: Acceptance ratio vs physical impairment threshold for the ARPANET network with $N_R = 20$.

Figure 3-24 shows the acceptance ratio vs physical impairment threshold for an the ARPANET network. As shown in the graph, BS_EISRA and BS_H1ISRA perform equally based on the acceptance ratio and algorithm BS_H2ISRA has a greater acceptance ratio than the other algorithms. This result may have been arisen from the fact that algorithm BS_H2ISRA encourages regenerator usage between backup paths whose primary paths are link-disjoint by temporarily making the weight of the links incident to the regenerator nodes of previously computed backup paths to zero.

The number of regenerators used by the accepted requests vs physical impairment threshold for an ARPANET network is shown in Figure 3-25. As can be seen from the figure regenerator number used by BS_H2ISRA is smaller than that of the other two algorithms even though its acceptance ratio is larger. This shows that the acceptance ratio of of the algorithm was larger for smaller physical impairment threshold values due to acceptance of requests whose backup paths can share regenerators. As expected BS_EISRA has a bit larger regenerator number than BS_H1ISRA.

Figure 3-26 shows the acceptance ratio vs total number of regenerators for an ARPANET network. The graph depicts that BS_EISRA and BS_H1ISRA perform equally based on the acceptance ratio, while algorithm BS_H2ISRA has a greater acceptance ratio due to

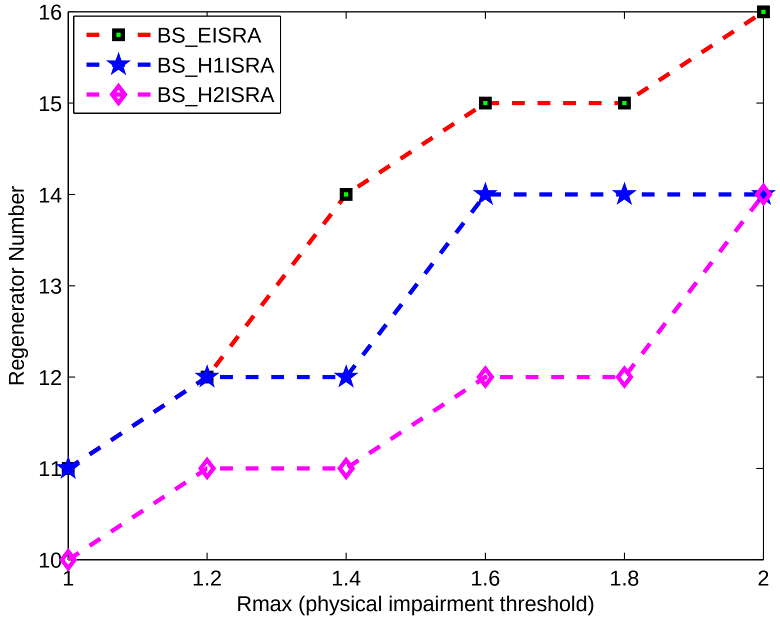

Figure 3-25: Path total regenerator number vs physical impairment threshold for the ARPANET network with $N_R = 20$.

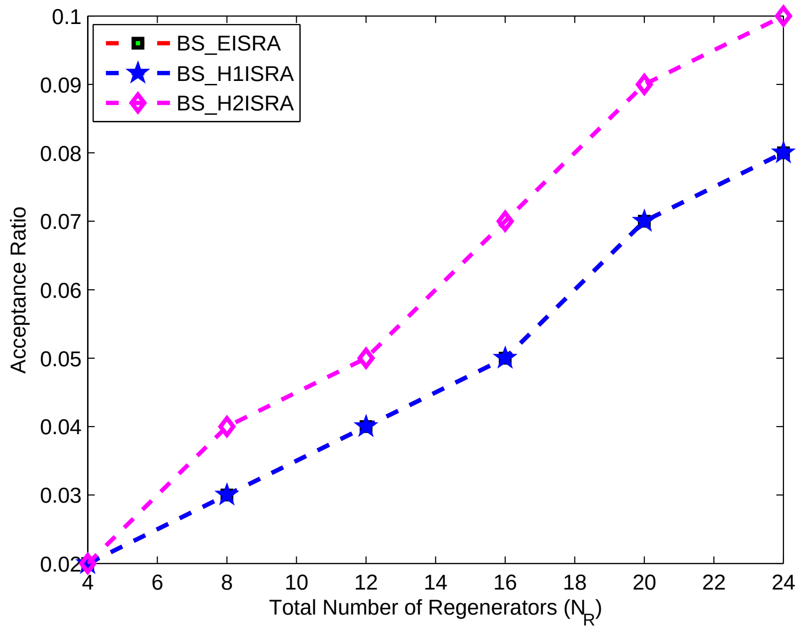

Figure 3-26: Acceptance Ratio vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

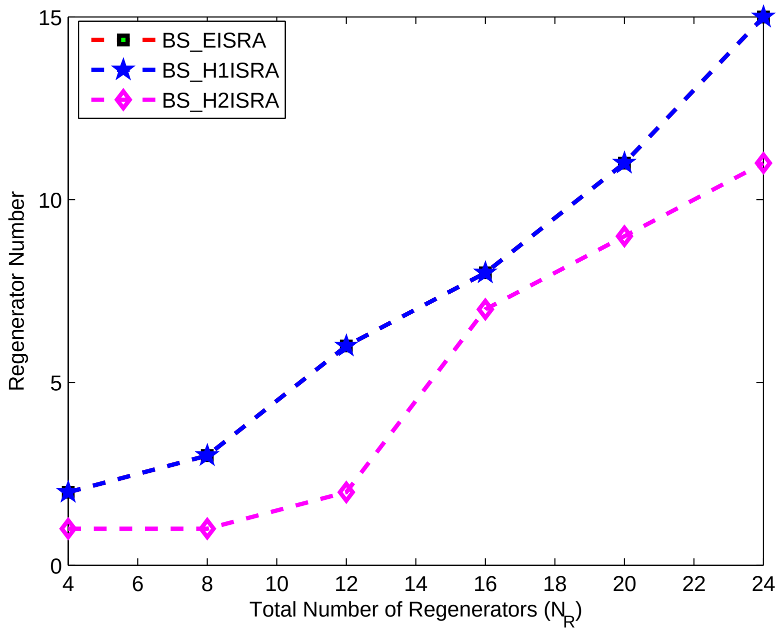

Figure 3-27: Path total regenerator number vs total number of regenerators for the ARPANET network with physical impairment threshold = 1.

the same reason discussed earlier.

Figure 3-27 shows the number of regenerators used by the accepted requests vs total number of regenerators for the ARPANET network. As can be seen from the figure, regenerator number used by BS_H2ISRA is smaller than that of the other two algorithms even though its acceptance ratio is larger due to the same reason discussed earlier. The other two algorithms, BS_EISRA and BS_H1ISRA, have the same regenerator number. The result also shows that the average regenerator number increases with the total number of regenerators in the network. This behavior is obtained due to the fact that increasing the number of regenerator nodes in a network for small physical impairment threshold leads to the acceptance of paths that require more regenerators, increasing the average regenerator number.

Chapter 4

Discussion, Conclusion and Future Work

4-1 Discussion

To solve the three survivable regenerator placement (SRP) problems with regenerator minimization objective, we have presented exact ILP and heuristic algorithms. The exact algorithms are formulated using network flow equations, while the heuristic algorithms are based on the concept of Suurballe's algorithm.

The algorithms given in chapter two solve the problem of: placing dedicated regenerators for the primary as well as the backup path of a request, placing shared regenerators between the primary and backup path of the same request and placing shared regenerators between backup paths of different requests whose primary paths are link-disjoint with each other, i.e., they do not fail at the same time during a link failure.

These algorithms are tested under different scenarios and their performance is measured in terms of the number of regenerators used along a path by varying the physical impairment threshold.

The exact algorithms attain the least number of regenerators compared to the heuristic algorithms at the expense of longer running time. Our experimental results further showed that, increasing the value of the physical impairment threshold reduces the number of regenerators required to admit a connection request. Furthermore, by sharing regenerators between primary and backup paths of a request as well as between backup paths of connections whose primary paths are link-disjoint, the number of regenerators required can be reduced.

To solve the dedicated impairment-aware survivable routing (DISR) problem we have proposed and implemented one exact and two heuristic algorithms. These algorithms were then modified to solve the other ISR sub-problems; regenerator sharing between the primary and backup path of a request, regenerator sharing between backup paths whose primary paths are link-disjoint, dedicated impairment-aware survivable routing under dynamic traffic and regenerator and wavelength sharing between backup paths whose primary paths are link-disjoint.

These algorithms are tested under different scenarios, and their performance is measured in terms of the acceptance ratio, the running time, and the number of regenerators used along a path. Since the algorithms provided show the same characteristics, except in the case of backup shared impairment-aware survivable routing, we will discuss only the dedicated and the backup shared algorithms.

The exact algorithm (D_EISRA) inherits some concepts of SAMCRA. The k-shortest path concept with unrestricted value of k_{max} and a modified version of the look-ahead concept are used. One of the heuristic algorithms (D_H1ISRA) is a modification of D_EISRA obtained by restricting the value of k_{max}. While the other heuristic algorithm (D_H2ISRA) is based on the concept of Suurballe's algorithm.

We have simulated the algorithms under different scenarios on different network topologies. While D_H2ISRA attains the least acceptance ratio, D_EISRA attains the largest acceptance ratio and D_H1ISRA follows it with a small margin. On the other hand, D_H2ISRA shows the best performance in terms of the running time, whereas D_EISRA shows the worst time performance because it performs a more exhaustive search to obtain the optimal solution which costs longer running time.

In the case of the algorithms provided for the backup shared impairment-aware survivable routing problem, BS_H1ISRA has the least acceptance ratio while BS_H2ISRA has the largest acceptance ratio followed by BS_EISRA. On the other hand, BS_H1ISRA shows the best performance in terms of running time, whereas BS_H2ISRA has the worst performance.

Our experimental results further showed that, increasing the number of regenerators in the network can improve network performance significantly in terms of acceptance ratio. The shared counterparts of these algorithms also show the same characteristics as described above.

4-2 Conclusion

In this thesis, we studied a translucent wavelength routed optical network architecture that effectively overcomes the signal quality degradation in a fully transparent network while using much less regenerators than a fully opaque network. We used the sparse regeneration technique which can significantly reduce the cost of the network.

The goal of this thesis is devising methods to solve survivable regenerator placement (SRP) problem and impairment-aware survivable routing (ISR) problem in translucent optical networks.

In Chapter 2, exact and heuristic algorithms to solve three SRP problems are presented with their simulation results and analysis of the results. While in Chapter 3, exact and heuristic algorithms that solve the different ISR problems are presented with their simulation results and analysis of the results.

The algorithms proposed in this thesis have their own advantages and disadvantages. Thus, the choice of a specific algorithm to solve a particular problem depends on the performance metric of our interest. The exact algorithms are preferable when optimality is the priority performance metric. However, the heuristic algorithms are the best choices when the running time is the priority.

4-3 Future Work

The results obtained in our work are encouraging and leave a broad avenue to explore for further research works. Therefore, we suggest the following works to be considered in future studies.

- In our study we only considered single link failures; but, in real world multiple links can fail at the same time. Therefore, the work can be extended to support multiple link failures.
- We also considered path protection mechanism which has higher recovery time; therefore considering segment protection mechanism which is a compromise between path based and link based protection mechanisms is another direction for future work.
- In the regenerator placement problems, we only considered placing a regenerator for a given request; but devising survivable regenerator placement algorithms based on different knowledge of future network traffic patterns can also be considered for future work.

Bibliography

[1] B. Ramamurthy, S. Yaragorla and X. Yang, "Translucent optical WDM networks for the next-generation backbone networks," in Proceedings of IEEE GLOBECOM 2001 Symposium on Optical and Photonic Communications (San Antonio, TX, 2001), 60-64.

[2] H. Zang and R. P. J. Huang, "Methodologies on Designing a Hybrid Shared-Mesh-Protected WDM Network with Sparse Wavelength Conversion and Regeneration," in Asia-Pacific Optical Communications Conference and Exhibition 2000, S. Xie, C. Qiao and Y. C. Chung, eds., Proc. SPIE 4910, 188-196 (2000).

[3] E. Yetginer and E. karasan, "Regenerator Placement and Traffic Engineering with Restoration in GMPLS Networks," Photonic Network Commun. 6, 134-149 (2003).

[4] M. Ali, "Shareability in Optical Networks: Beyond Bandwidth Optimization," IEEE Commun. Mag. 42, 11-15 (2004).

[5] X. Yang, L. Shen, and B. Ramamurthy, "Survivable Lightpath Provisioning in WDM Mesh Networks under Shared Path Protection and Signal Quality Constraint," J. Lightwave Technol. 23, 1556-1567 (2005).

[6] Hui Zang, Renxiang Huang, and James Pan, "Methodologies on Designing a Hybrid Shared-Mesh Protected WDM Network with Sparse Wavelength Conversion and Regeneration," Proc. of SPIE, vol. 4910, (Sep. 2000), pp. 188-196.

[7] Bhandari R., "Optimal diverse routing in telecommunication fiber networks", Proceedings of IEEE INFOCOM '94. Toronto, Ontario, Canada, vol. 3, June, 1994; 1498-1508.

[8] J.W. Suurballe, "Disjoint paths in a network", Networks 1974; 4:125-145.

[9] Y. Guo, F.A. Kuipers, and P. Van Mieghem, "A Link Disjoint Paths Algorithm for Reliable QoS Routing," Int. Journal of Communication Systems, vol. 16, no. 9, 2003, pp. 779-798.

[10] Xi Yangz and Byrav Ramamurthy, "Sparse Regeneration in Translucent Wavelength-Routed Optical Networks: Architecture, Network Design and wavelength Routing," Photonic Network Communications, Vol. 10, No. 1. (1 July 2005), pp. 39-53.

[11] Si Chen, Ivana Ljubic, and S. Raghavan, "The Generalized Regenerator Location Problem", International Network Optimization Conference, April 26-29, 2009, Pisa, Italy.

[12] Michele Flammini, Alberto Marchetti Spaccamela, Gianpiero Monaco, Luca Moscardelli and Shmuel Zaks, "On the Complexity of the Regenerator Placement Problem in Optical Networks," Proc. of SPAA '09, pp.154-162, 2009.

[13] Donald B. Johnson, "Efficient Algorithms for Shortest Paths in Sparse Networks," Journal of the ACM (JACM), vol.24, no.1, pp. 1-13, Jan. 1977.

[14] Kostas Katrinis, Anna Tzanakaki and George Markidis, "Impairment Aware WDM Network Dimensioning with Optimized Regenerator Placement," National Fiber Optic Engineers Conference (NFOEC), March 22, 2009.

[15] S.W. Kim, S.W. Seo and S.C. Kim, "Regenerator Placement Algorithms for Connection Establishment in all Optical Networks," Proc. IEEE Communications, vol. 148, Issue 1, pp 25-30, Feb. 2001.

[16] S. Pachnicke, T. Paschenda, and P. M. Krummrich, "Physical Impairment Based Regenerator Placement and Routing in Translucent Optical Networks," Optical fiber communication/National Fiber Optic Engineers Conference, OFC/NFOEK 2008, pp. 1-3, Feb. 2008.

[17] G. Shen, Wayne D. Grover, Tee Hiang Cheng and Sanjay K. Bose, "Sparse placement of electronic switching nodes for low blocking in translucent Optical Networks," OSA Journal of Optical Networking, vol. 1, no. 12, pp. 424-41, Dec. 2002.

[18] A. Todimala, B. Ramamurthy, "A dynamic partitioning protection routing technique in WDM networks", Cluster Computing, vol. 7, no. 3, pp. 259-269, July 2004.

[19] C.V. Saradhi, L.K. Wei and M. Gurusamy, "Provisioning fault-tolerant scheduled lightpath demands in WDM mesh networks", International Conference on Broadband Networks, pp. 150-159, October 2004.

[20] G. Mohan, C.S.R. Murthy, and A. Somani, "Efficient algorithms for routing dependable connections in WDM optical networks," IEEE/ACM Transactions on Networking, vol. 9, pp. 553-566, Oct. 2001.

[21] H. Zang, C. Ou, B. Mukherjee, “Path-Protection Routing and. Wavelength Assignment (RWA) in WDM Mesh Networks under. Duct-layer Constraints”, IEEE/ACM Transactions on Networking, vol. 11, no. 2, pp. 248-258 2003.

[22] M. Sridharan, M.V. Salapaka, and A.K. Somani, “A Practical Approach to Operating Survivable WDM Networks,” IEEE Journal on Selected Areas in Communications, vol. 20, no.1, January 2002.

[23] J. Harmatos and P. Laborczi “Dynamic routing and wavelength assignment in survivable WDM networks,” Photonic Network Communications, vol. 4, no. 3/4, , pp. 357-376, July 2002.

[24] H. Zang, J.P. Jue, and B. Mukherjee, “A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks,” SPIE Optical Networks Magazine, vol.1, no.1, pp.47-60, Jan.2000.

[25] A. Fumagalli and L. Valcarenghi, “IP restoration vs. WDM protection: Is There an Optimal Choice?,” IEEE Network, vol. 14, no. 6, pp. 34-41,. Nov./Dec. 2000.

[26] M. Cuna and M. Mellia, “Comparison of protection cost at IP or WDM layer,” Workshop on High Performance Switching and Routing, pp. 287-293, 2006.

[27] M. Ali Ezzahdi, S. Al Zahr, M. Koubaa, N. Puech, and M. Gagnaire, “LERP: a Quality of Transmission Dependent Heuristic for Routing andWavelength Assignment in Hybrid WDM Networks,” Proc. of ICCCN, pp. 125-136, October 2006.

[28] R. Cardillo, V. Curri, and M. Mellia., “Considering Transmission Impairments in Wavelength Routed Networks,” Proc. of ONDM, pp. 421-429, February 2005.

[29] H. Yurong, J.P. 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.

[30] J. He, M. Brandt-Pearce, Y. Pointurier, and S. Subramaniam, “QoT-Aware Routing in Impairment-Constrained Optical Networks,” Proc. of IEEE GLOBECOM, pp. 2269-2274, November 2007.

[31] C. Siva Ram Murthy, and G. Mohan, “WDM Optical Networks: Concepts, Design, and Algorithms”, Prentice Hall PTR, 2002.

[32] Si Chen, Ivana Ljubic and S. Raghavan, “The Regenerator Location Problem”, Networks 2008.

[33] S. Pachnicke, T. Paschenda, and P.M. Krummrich, “Physical Impairment Based Regenerator Placement and Routing in Translucent Optical Networks,” Proc. of the Optical Fiber communication/National Fiber Optic Engineers Conference, OFC/NFOEC 2008, pp. 1-3, Feb. 2008.

[34] S. Azodolmolky, M. Klinkowski, Eva Marin, Davide Careglio, J. S. Pareta, 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.

[35] C. Qiao, D. Datta, G. Ellinas, A. Gladisch, and E. Modiano (Eds.), “WDM-based network architectures,” IEEE Journal on Selected Areas in Communications, vol. 20, no. 1, Jan. 2002.

[36] K. Sivalingam and S. Subramaniam, Eds., “Emerging Optical Network Technologies. Boston, MA: Springer Publishers, 2004.

[37] M. Sivakumar, R. Shenai, and K.M. Sivalingam, “Protection and restoration for optical WDM networks: A survey,” in Emerging Optical Network Technologies, K. Sivalingam and S. Subramaniam, Eds. Springer Publishers, 2004.

[38] D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He, “On Finding Disjoint Paths in Single and Dual Link Cost Networks,” in Proc. IEEE INFOCOM, Hong Kong, Mar. 2004.

[39] P.H. Ho, J. Topolcai, and H.T. Mouftah, “On achieving optimal survivable routing for shared protection in survivable next-generation internet,” IEEE Transactions on Reliability, vol. 53, no. 2, pp,216-225 June 2004.

[40] C.S.R Murthy and M. Gurusamy, “Lightpath restoration in WDM optical networks”, Network, IEEE, Vol. 14, no. 6, pp. 24-32, Nov./Dec. 2000.

[41] M. Patel, R. Chandrasekaran, and S. Venkatesan, “Improved Quasi-Path Restoration in Mesh Networks,” IEEE/ACM Transactions on Networking, vol. 16, no. 1, Feb. 2008.

[42] A. Birman, “Computing approximate blocking probabilities for a class of all-optical networks,” in Proc. IEEE INFOCOM, Apr. 1995, pp. 651-658.

[43] R.R. Iraschko, W.D. Grover, and M.H. MacGregor, “A distributed real time path restoration protocol with performance close to centralized multicommodity max flow,” International Workshop on Design of Reliable Communication Networks (DRCN), May 1998.

[44] S. Venkatesan, M. Patel, and N. Mittal, “A distributed algorithm for path restoration in circuit switched communication networks,” 24th IEEE Symposium Reliable Distributed Systems, pp. 226-235, Oct. 2005.

[45] G. Mohan and A.K. Somani, “Routing dependable connections with specified failure restoration guarantees in WDM networks,” INFOCOM 2000.

[46] P. Van Mieghem and F.A. Kuipers, "Concepts of Exact Quality of Service Algorithms," IEEE/ACM Transaction on Networking, vol. 12, no. 5, pp. 851-864, October 2004.

[47] C.R. Giles and E. Desurvivre, "Propagation of Signal and Noise in Concatenated Erbium-Doped Fiber Optical Amplifiers," IEEE/OSA Journal of Lightwave Technology, vol. 9, no. 2, Feb. 1991.

[48] J. Strand, A.L. Chiu, and R. Tkach, "Issues for Routing in the Optical Layer," IEEE Communications Magazine, vol. 39, no. 2, Feb.2001.

Appendix A

Additional Results

A-1 Survivable Regenerator Placement Problem

A-1-1 Multiple Physical Impairments Heuristic Algorithms

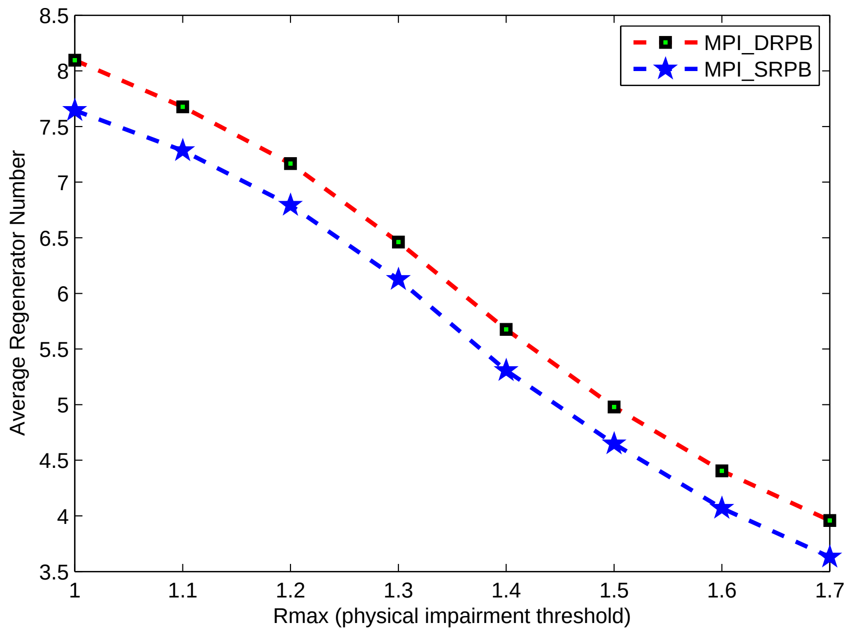

Figure A-1: Average regenerator number vs physical impairment threshold for lattice network with $N = 49$ (MPI_DRPB vs MPI_SRPB).

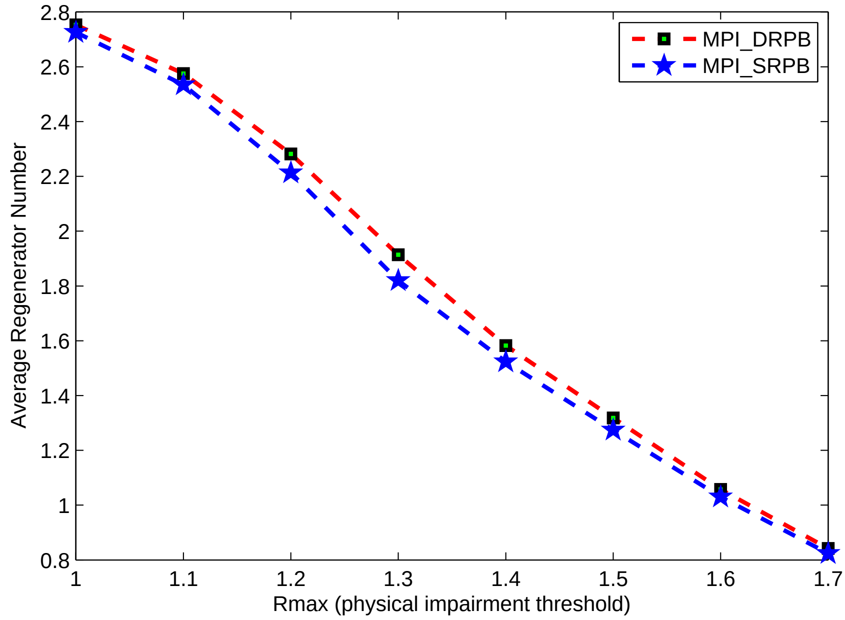

Figure A-2: Average regenerator number vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (MPI_DRPB vs MPI_SRPB).

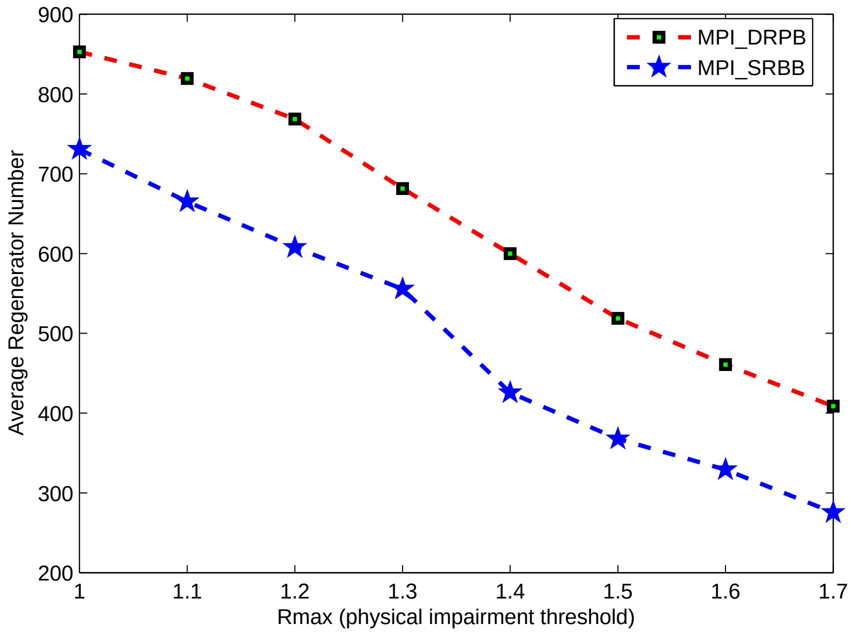

Figure A-3: Average regenerator number vs physical impairment threshold for lattice network with $N = 49$ (MPI_DRPB vs MPI_SRBB).

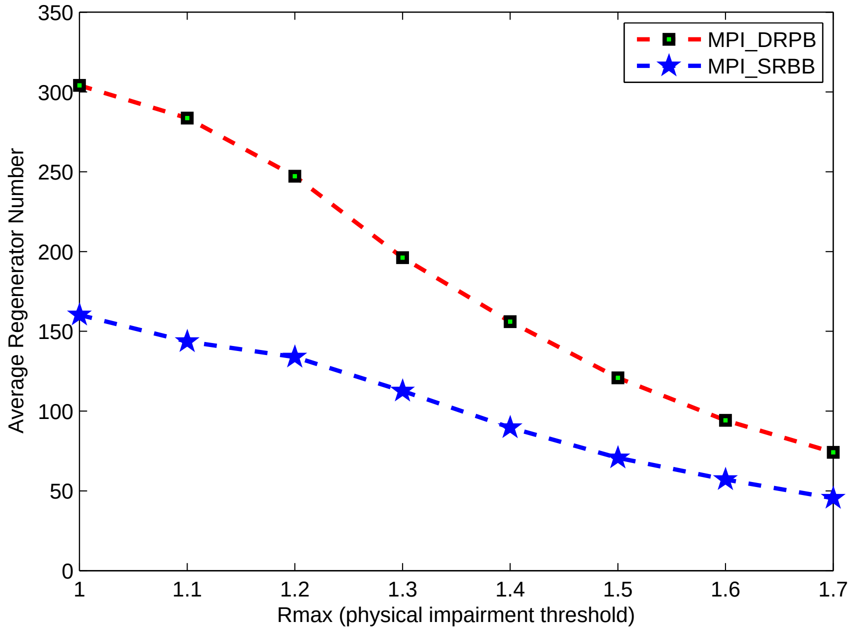

Figure A-4: Average regenerator number vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (MPI_DRPB vs MPI_SRBB).

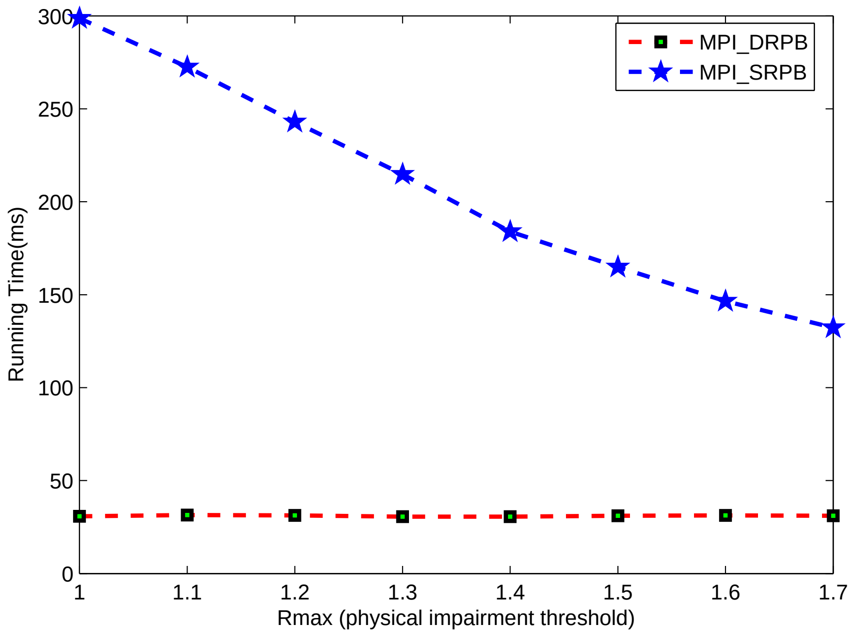

Figure A-5: Running time vs physical impairment threshold for lattice network with $N = 49$ (MPI_DRPB vs MPI_SRPB).

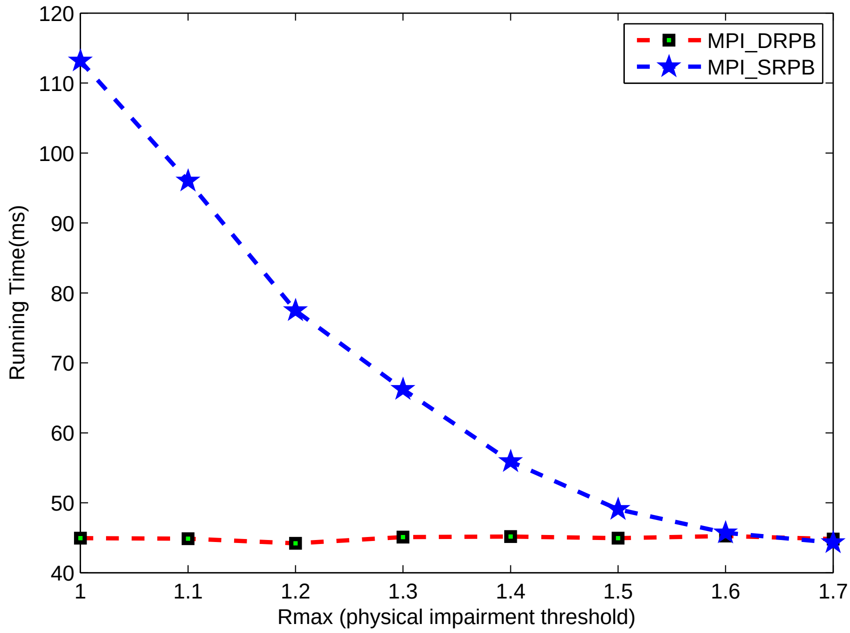

Figure A-6: Running time vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (MPI_DRPB vs MPI_SRPB).

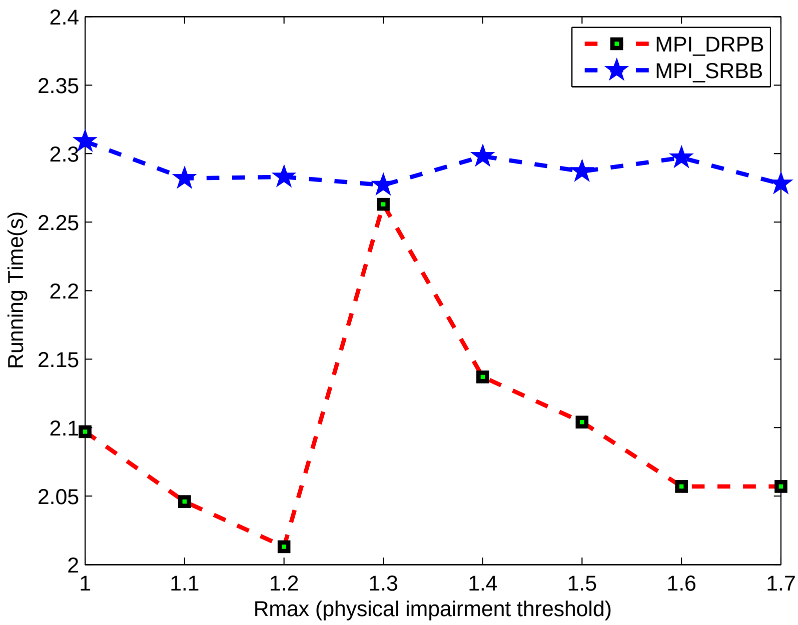

Figure A-7: Running time vs physical impairment threshold for lattice network with $N = 49$ (MPI_DRPB vs MPI_SRBB).

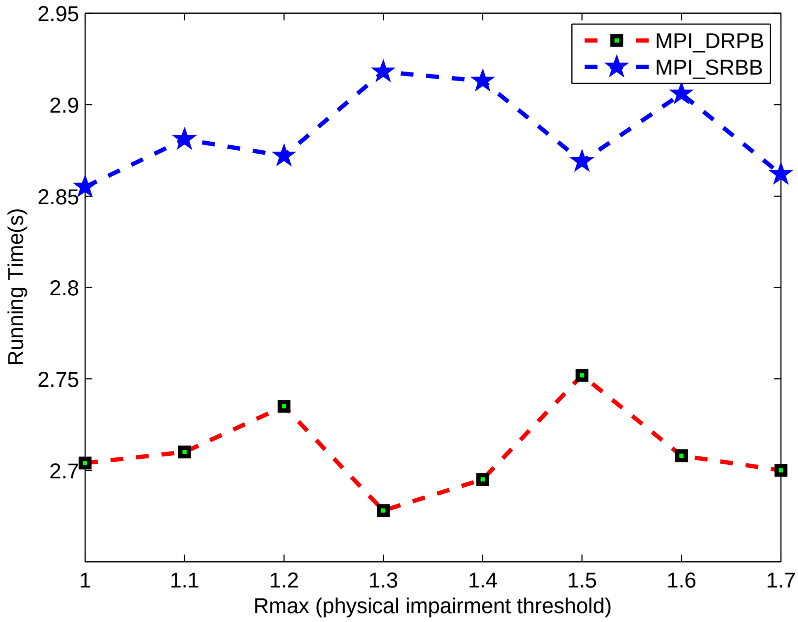

Figure A-8: Running time vs physical impairment threshold for random network with $N = 49$ and $\rho = 0.2$ (MPI_DRPB vs MPI_SRBB).

A-2 Impairment-aware Survivable Routing Problem

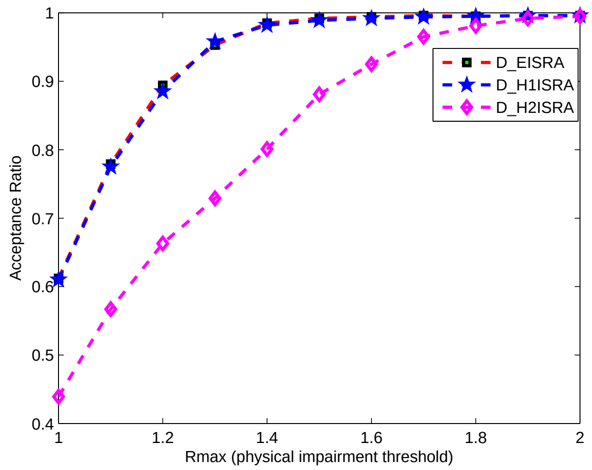

Figure A-9: Acceptance ratio vs physical impairment threshold for NSFNET network ($N_R = 6$).

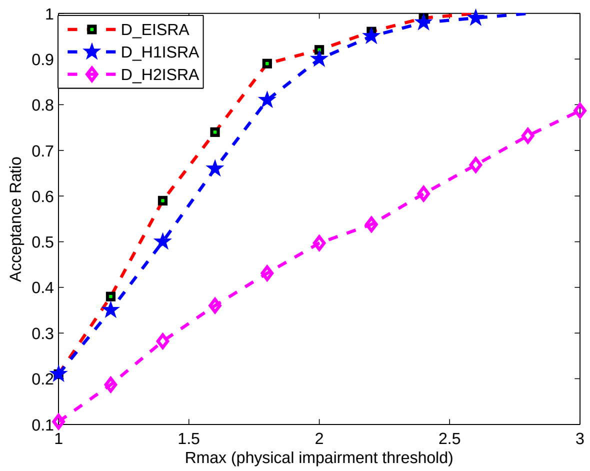

Figure A-10: Acceptance ratio vs physical impairment threshold for lattice network with $N = 49$ and $N_R = 12$.

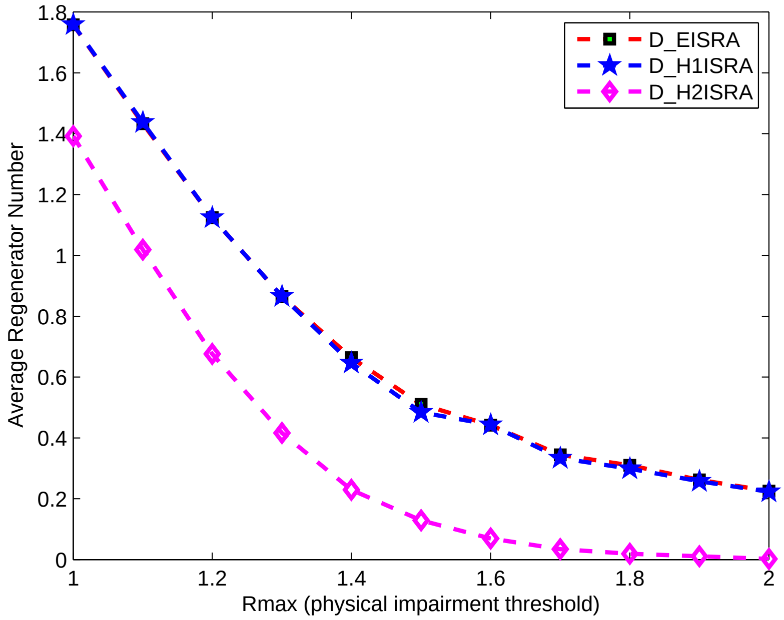

Figure A-11: Average regenerator number vs physical impairment threshold for NSFNET network with $N_R = 6$.

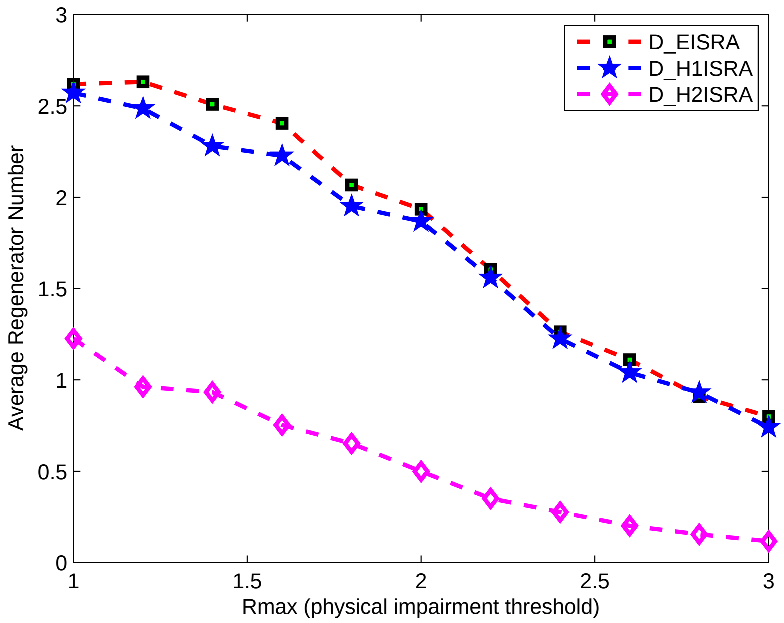

Figure A-12: Average regenerator number vs physical impairment threshold for lattice network with $N = 49$ and $N_R = 12$.

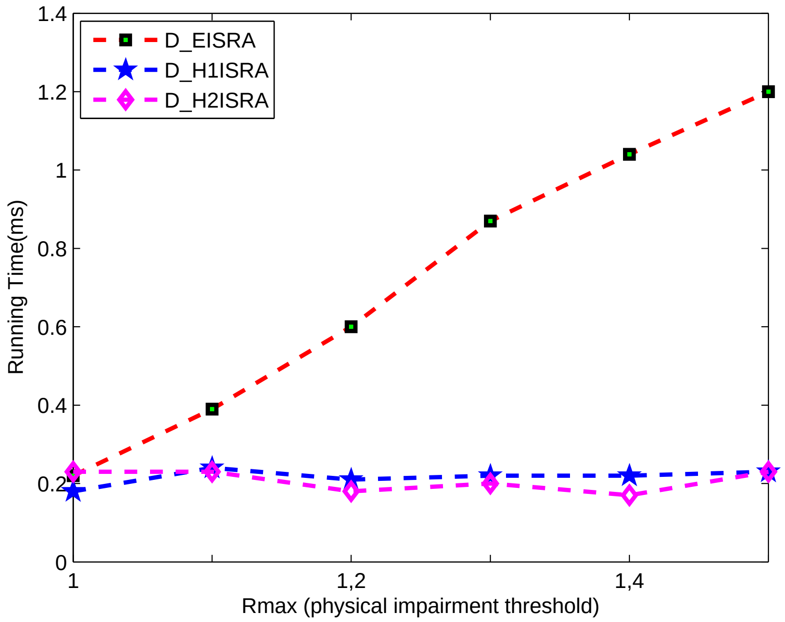

Figure A-13: Running time vs physical impairment threshold for NSFNET network with $N_R = 6$.

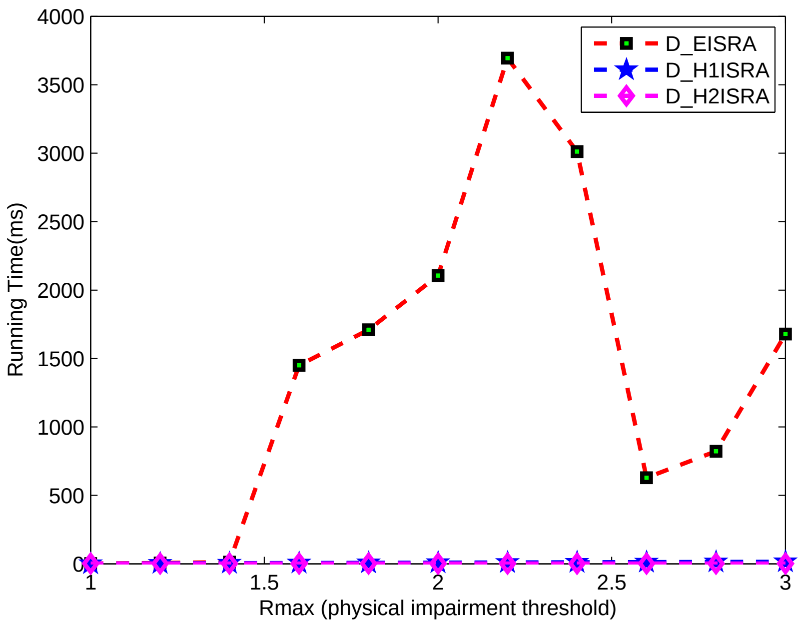

Figure A-14: Running time vs physical impairment threshold for lattice network with $N = 49$ and $N_R = 12$.

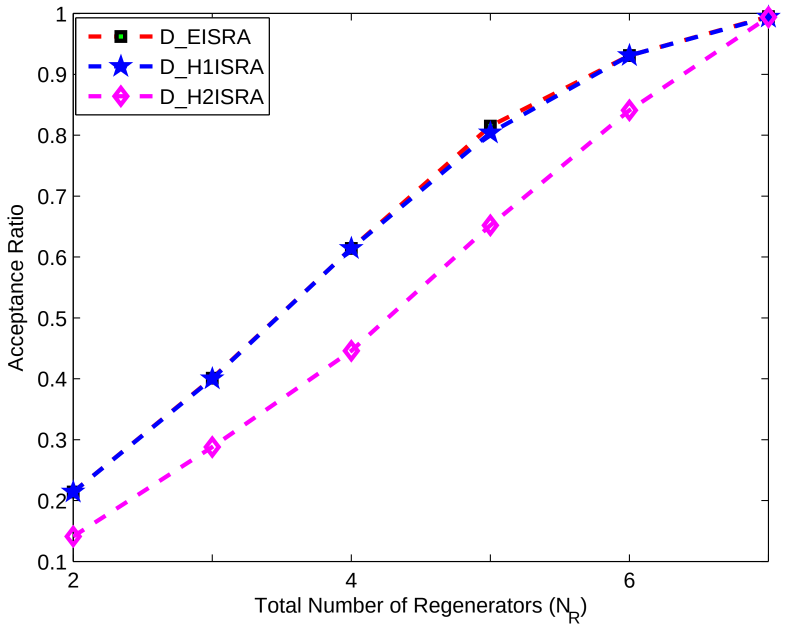

Figure A-15: Acceptance Ratio vs total number of regenerators for NSFNET network with physical impairment threshold $= 1$.

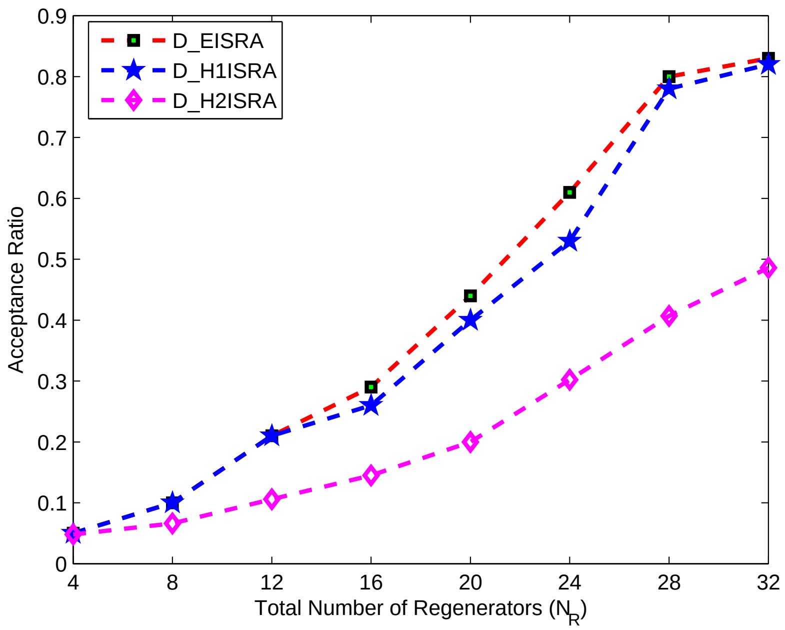

Figure A-16: Acceptance Ratio vs total number of regenerators for lattice network with $N = 49$ and physical impairment threshold $= 1$.

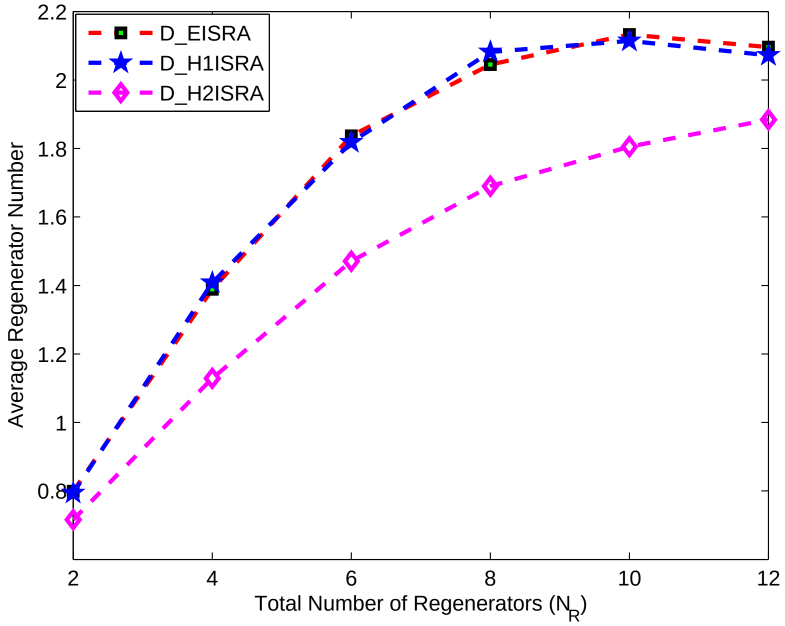

Figure A-17: Average regenerator number vs total number of regenerators for NSFNET network with physical impairment threshold $= 1$.

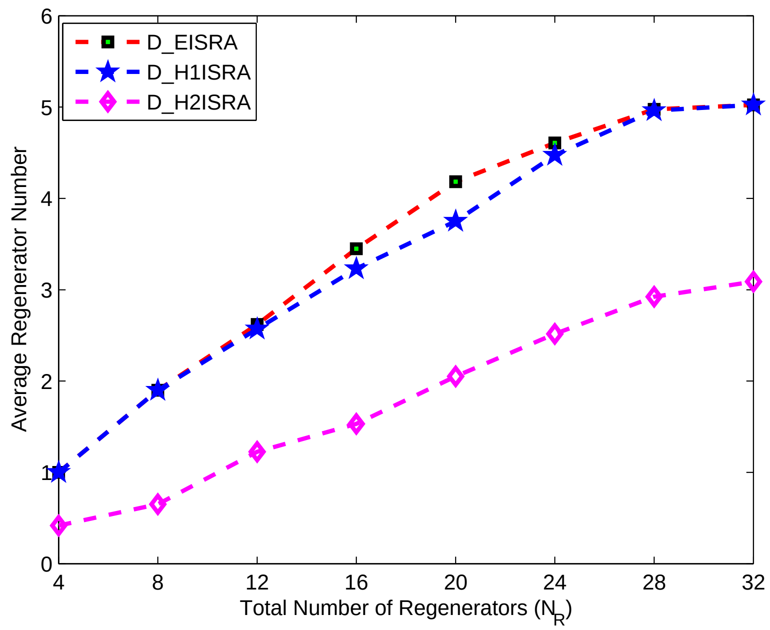

Figure A-18: Average regenerator number vs total number of regenerators for lattice network with $N = 49$ and physical impairment threshold $= 1$.

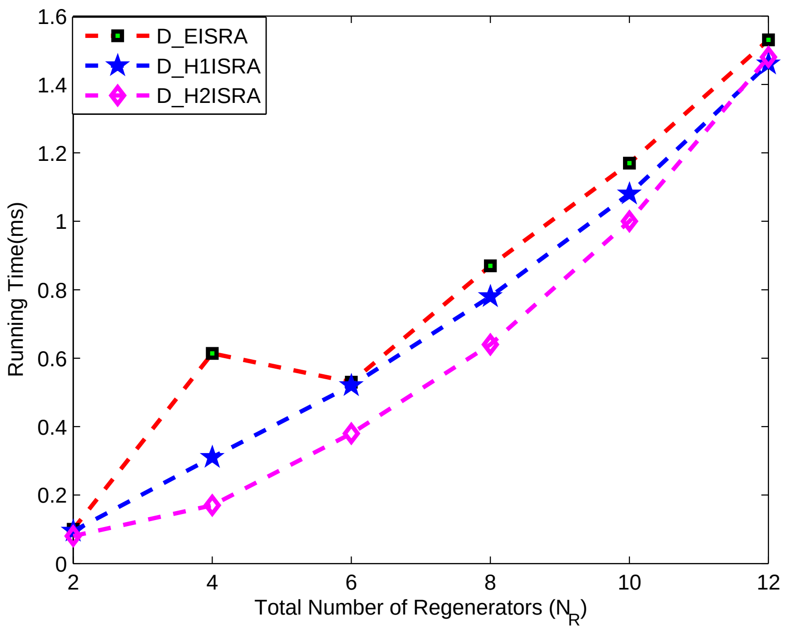

Figure A-19: Running time vs total number of regenerators for NSFNET network with physical impairment threshold $= 1$.

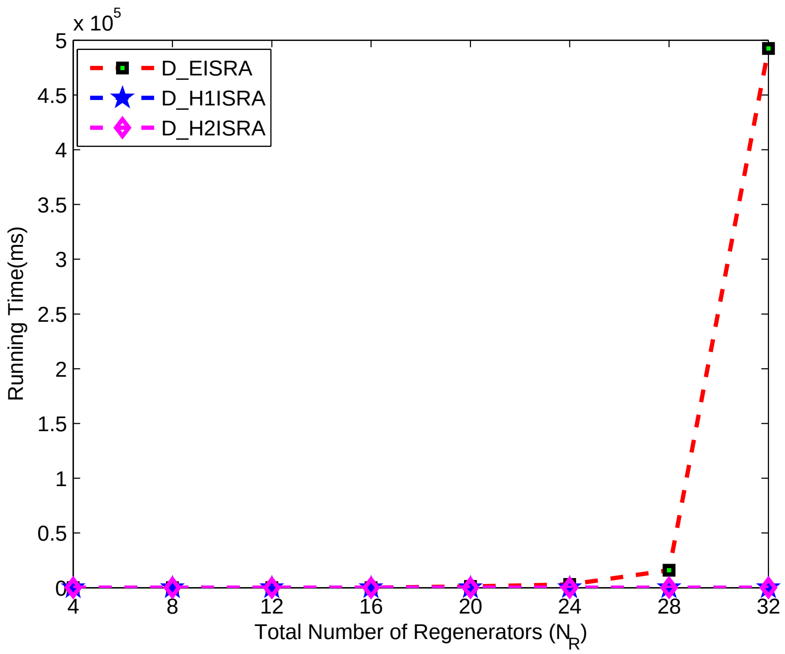

Figure A-20: Running time vs total number of regenerators for lattice network with $N = 49$ and physical impairment threshold $= 1$.