

SUPPORTING DIFFERENTIATED CLASSES OF RESILIENCE IN MULTILAYER NETWORKS

by

Abdulaziz Saleh A. Alashaikh

B.S. in EE, King Abdulaziz University, Saudi Arabia, 2004

MST, University of Pittsburgh, Pittsburgh, PA, 2011

Submitted to the Graduate Faculty of
the School of Information Science in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH
SCHOOL OF INFORMATION SCIENCE

This dissertation was presented

by

Abdulaziz Saleh A. Alashaikh

It was defended on

July 20, 2017

and approved by

David Tipper, Professor, Telecommunications and Networking Program, University of
Pittsburgh

Martin B.H. Weiss, Professor, Telecommunications and Networking Program, University of
Pittsburgh

Prashant Krishnamurthy, Associate Professor, Telecommunications and Networking
Program, University of Pittsburgh

Teresa Gomes, Assistant Professor, Department of Electrical and Computers Engineering,
University of Coimbra

Dissertation Director: David Tipper, Professor, Telecommunications and Networking
Program, University of Pittsburgh

Copyright © by Abdulaziz Saleh A. Alashaikh
2017

ABSTRACT

**SUPPORTING DIFFERENTIATED CLASSES OF RESILIENCE IN
MULTILAYER NETWORKS**

Abdulaziz Saleh A. Alashaikh, PhD

University of Pittsburgh, 2017

Services provided over telecommunications networks typically have different resilience requirements and networks need to be able to support different levels of resilience in an efficient manner. This dissertation investigates the problem of supporting differentiated classes of resilience in multilayer networks, including the most stringent resilience class required by critical services. We incorporate an innovative technique of embedding a subnetwork, termed the spine, with comparatively higher availability values at the physical layer. The spine lays a foundation for differentiation between multiple classes of flows that can be leveraged to achieve both high resilience and differentiation. The aim of this research is mainly to explore, design, and evaluate the proposed spine concept model in multilayer networks. The dissertation has four major parts. First, we explore the spine concept through numerical analysis of simple topologies illustrating the potential benefits and the cost considerations of the spine. We develop heuristics algorithms to find suitable spines for a network based on the structural properties of the network topology. Second, an optimization problem is formulated to determine the spine. The problem encompasses estimates of link availability improvements, associated costs, and a total budget. Third, we propose a crosslayer mapping and spine-aware routing design problem with protection given mainly at the lower layer. The problem is designed to transfer lower layer differentiation capability to the upper layer network and flows. We provide two joint routing-mapping optimization formulations and evaluate their performance in a multilayer scenario. Fourth, the joint routing-mapping

problem is redesigned with protection given in the upper network layer instead. This will create two isolated logical networks; one mapped to the spine and the other is mapped freely on the network. Flows are assigned a path or path-pair based on their class of resilience. This approach can provide more routing options yielding different availability levels. The joint routing-mapping design problems are formulated as Integer Linear Programming (ILP) models. The goal is to achieve a wider range of availability values across layers and high availability levels for mission-critical services without the need to use higher order protection configurations. The proposed models are evaluated with extensive numerical results using real network topologies.

Keywords: Differentiated Resilience, Availability, Multilayer Networks Design.

TABLE OF CONTENTS

PREFACE	xv
1.0 INTRODUCTION	1
1.1 Differentiated Classes of Service	1
1.2 Motivation	4
1.3 Objective	5
1.4 Organization	5
2.0 BACKGROUND	6
2.1 Multilayer Networks	6
2.1.1 IP and WDM Technology	6
2.1.2 Crosslayer Mapping	8
2.2 Quality of Resilience (QoR)	10
2.2.1 Availability Model	10
2.2.2 Classes of QoR	14
2.2.3 Failures in Communication Networks	15
2.2.4 Techniques to Improve Equipment Availability	17
2.3 Survivable Networks	18
2.3.1 Recovery Mechanisms	19
2.3.2 Survivable Network Design (SND)	21
2.3.3 Differentiated Classes of Survivability	22
3.0 LITERATURE REVIEW	25
3.1 Routing Algorithms	25
3.1.1 Optimization Problem Approach	25

3.1.2	Alternate Path Method	27
3.1.3	Weighted Routing Approach	28
3.2	Availability-Aware Network Design	29
3.2.1	Alternating Recovery Mechanisms	29
3.2.1.1	Shared Protection	30
3.2.1.2	Restoration	32
3.2.1.3	Multipath Routing (MP)	33
3.2.1.4	P-cycles	34
3.2.2	A Time Perspective Recovery	35
3.2.3	Overall Network Availability	35
3.3	Availability in Multilayer Networks	37
3.3.1	Mapping Link Availability	37
3.3.2	Types of crosslayer problems	38
3.3.3	The case of overlay networks	39
3.4	Summary	40
4.0	RESEARCH OVERVIEW	41
4.1	Overview	41
4.2	Thesis Statement	42
4.3	Scope and Assumptions	42
4.4	Methodology	43
4.5	List of Publications	44
4.6	Contribution	44
5.0	THE SPINE CONCEPT	46
5.1	Definition	46
5.1.1	The Spine Model	48
5.1.2	Implementing Heterogenous Link Availabilities	48
5.2	Exploring the spine	49
5.2.1	Exploiting the Heterogeneity of Link Availability	50
5.2.2	Effect of Network Size	53
5.2.3	The Layout of The Spine	54

5.3	Exploring <i>Spine</i> Selection	56
5.3.1	Notation	57
5.3.2	Generating Candidate Spines	59
5.3.3	Numerical Results	60
5.4	Considering All Spanning Trees in Determining the Spine	64
5.4.1	Generating All Spanning Trees	64
5.4.2	Numerical Results	65
5.4.3	Sensitivity Analysis	69
5.4.4	Monetary Cost and Implementation Issues	72
5.5	The Spine Link Selection Design Problem	75
5.5.1	Problem Statement	75
5.5.2	Notation	77
5.5.3	Incremental Link Availability Model	77
5.5.4	Optimization Model Formulations	79
5.5.5	Numerical Study	84
5.5.5.1	Cost Functions	84
5.5.5.2	Results	87
5.6	Summary	103
6.0	THE JOINT CROSSLAYER ROUTING-MAPPING DESIGN PROBLEM	105
6.1	Introduction	105
6.2	Protection Configuration	106
6.3	Protection at Lower Layer	109
6.3.1	Notation	109
6.3.2	Network Model	111
6.3.3	Formulation	111
6.3.3.1	Problem I: LP for duplicated logical links	111
6.3.3.2	Problem Type II: MILP for partitioned logical network	113
6.3.4	Numerical Study	114
6.3.4.1	Scenarios	114

6.3.4.2 Results	117
6.4 Protection at Upper Layer	125
6.4.1 Notation	127
6.4.2 Problem Statement	128
6.4.3 Problem Formulation	129
6.4.3.1 Logical Subnetworks Design Problem Formulation	129
6.4.3.2 Routing on Logical Subnetworks Problem	132
6.4.4 Computational Complexity	133
6.4.5 Numerical Study	134
6.4.5.1 Scenarios	134
6.4.5.2 Results	134
6.5 Summary	150
7.0 CONCLUSIONS AND SUMMARY	151
7.1 Contribution	151
7.2 Limitations and Future Research	154
BIBLIOGRAPHY	159

LIST OF TABLES

2.1	An example of recovery classes.	23
5.1	Effect of varying ϵ on A_S and Downtime.	51
5.2	Test Network Topology Data.	60
5.3	Numerical Results for Heuristics.	61
5.4	Number of spanning trees for sample networks.	65
5.5	Results considering all spanning trees.	66
5.6	Structural properties of the resulted spines.	88
6.1	Possible protection configurations for a multilayer network with a spine. . . .	108
6.2	Test Networks.	114
6.3	Average resource use for the three spines in all scenarios.	118
6.4	Average and maximum logical link expected downtime.	121
6.5	Average and maximum logical flows expected downtime.	122
6.6	A comparison of the resource use between the spine and the baseline model. .	123
6.7	Average and maximum flow expected downtime for the baseline model. . . .	124
6.8	Traffic ratio scenarios.	125
6.9	QoR class-1 average resource use for the three spines in all scenarios.	136
6.10	Total resources used for QoR class-1 in the two protection configurations (Bot- tom Layer results shown in Table 6.3.	137
6.11	Resource use for unconstrained routing on logical subnetworks X and Y for spine S_2	139
6.12	A comparison of the resource use for single unprotected paths for spine S_2 between the two configurations.	140

6.13 Average and maximum logical link expected downtime.	141
6.14 Average and maximum logical flows expected downtime.	142
6.15 A comparison of the resource use between the spine and the baseline model. .	147
6.16 The expected downtime of the baseline model (<i>i.e.</i> , no spine model).	148
6.17 Traffic ratio scenarios.	148

LIST OF FIGURES

2.1	An example of backhaul mapping.	9
2.2	An example of survivable mapping.	9
2.3	The Markov state model for availability.	11
2.4	Typical Components Availability.	16
2.5	Classification of recovery mechanisms.	19
3.1	Availability comparison of basic recovery options.	30
3.2	An example of self-protecting multipath routing.	33
3.3	An example of a dual-failure scenario with p-cycle	34
4.1	An example of a spine.	41
5.1	A plot shows that strengthen the strongest link in parallel configuration achieves the best overall end-to-end availability.	47
5.2	Full mesh network (thicker red lines denote spine).	50
5.3	Average Downtime corresponding to A_S versus a and Δ	52
5.4	The effect of differential links availabilities on different sizes networks.	54
5.5	An example of a 5 nodes network with two spine designs.	55
5.6	Average Downtime corresponding to A_S versus a and Δ for the two 5 nodes examples.	56
5.7	The MSTs obtained by heuristics for Polska network – red/thicker lines rep- resent the spine.	62
5.8	A_S^{WP} , A_S & min- A_f calculated over all spanning trees for Polska Network. . .	67
5.9	A_S^{WP} , A_S & min- A_f calculated over all spanning trees for NSF Network. . .	67
5.10	A_S^{WP} , A_S & min- A_f calculated over all spanning trees for Spain Network. . .	68

5.11 A_S^{WP} , A_S & min- A_f calculated over all spanning trees for EPAN16 Network.	68
5.12 The effect of varying a in Polska network.	69
5.13 The effect of varying Δ in Polska network.	70
5.14 Average Downtime corresponding to A_S for all STs at different Δ values.	70
5.15 Scatterplot of Polska STs measures with homogenous versus heterogenous link availability.	72
5.16 Scatterplot of NSF STs measures with homogenous versus heterogenous link availability.	72
5.17 Comparison between total cost of the spine design and non-spine design.	76
5.18 An example of the incremental link availability model for the Polska network.	79
5.19 The optimal spine with $a_{ij}^1 = 0.99$ and $a_{ij}^2 = 0.999$ (in green) compared to heuristic spines and all spanning trees of the Polska Network.	84
5.20 Improved link availability versus cost for three links from the Polska network example.	86
5.21 CDF of the link improvement costs.	86
5.22 Spines obtained for the Polska network using the three cost functions.	90
5.23 The corresponding link downtime/year and versus link length for the spines obtained for the Polska network.	91
5.24 Node Degree for the resulted Polska spines.	92
5.25 Spines obtained for the Spain network using the three cost functions.	93
5.26 The corresponding link downtime/year and versus link length for the spines obtained for the Spain network.	94
5.27 Node Degree for the resulted Spain spines.	95
5.28 Spines obtained for the Italia14 network using the three cost functions.	96
5.29 The corresponding link downtime/year and versus link length for the spines obtained for the Italia14 network.	97
5.30 Node Degree for the resulted italia14 spines.	98
5.31 Average expected WP and end-to-end flow downtime/year versus cost for the spines obtained for the Polska network.	99

5.32 Average expected WP and end-to-end flow downtime/year versus cost for the spines obtained for the Spain network.	100
5.33 Average expected WP and end-to-end flow downtime/year versus cost for the spines obtained for the Italia14 network.	101
5.34 The range of paths expected downtime in the spines of the Polska network. . .	102
5.35 The range of paths expected downtime in the spines of the Spain network. . .	102
5.36 The range of paths expected downtime in the spines of the Italia14 network. .	103
6.1 Protection at bottom layer.	106
6.2 Protection at top layer.	106
6.3 Illustration of 3 classes with different configurations with protection, if appli- cable, given on top layer.	107
6.4 The layout of the spines used for the test networks.	115
6.5 Polska network with different spines and logical networks.	117
6.6 Percentage of increase in total resources compared to the baseline model for the Polska network.	126
6.7 Downtime comparison between the two protection configurations on the Polska network.	143
6.8 Downtime comparison between the two protection configurations on the NSF network.	144
6.9 Downtime comparison between the two protection configurations on the Spain network.	144
6.10 Downtime comparison between the two protection configurations on the Italia14 network.	145
6.11 Downtime comparison between the two protection configurations on the EPAN network.	145
6.12 Percentage of increase in total resources compared to the baseline model in the Polska network and for different traffic ratio mix.	149

PREFACE

First, I would like to thank and express my deepest and sincerest gratitude to my advisor Dr. David W. Tipper for his support, guidance, and all the help he has given me over the years of my PhD study. His vision, knowledge, expertise, humility and patience have been the best inspiration, motivation, and the bond of trust for his students. It was a great privilege and honor to be one of his students. I also would like to extend my thanks to Dr. Martin B.H. Weiss, Dr. Prashant Krishnamurthy, and Dr. Teresa Gomez for serving on my committee and providing me with their thoughtful inputs and comments that enriched my research. I would like to say thanks to all my colleagues, and to the faculty members and staff of the information science school. In addition, I would like to acknowledge the Wisconsin Institute for Discovery at the University of Wisconsin in Madison and Gurobi Optimization for the generous access they offer to public and academic students. More importantly, I would like to acknowledge the generous scholarship received from the Saudi Arabian Cultural Mission (SACM).

I would like to express my deepest thanks and love to all my family: my mother *Sara* for her endless prayers and continuous love and support I received from her since birth. My lovely wife *Laila* for her endless love, support, encouragement, and patience throughout the course of my study. She was the closest friend and the one who always believes in me. I owe her a lot and all thanks will not suffice. I want to thank our son *Saleh*, whom I was overly busy to give him the parental time he deserves. My grandmother and my mother in law whom were eagerly waiting for my success and return. Thanks to all my family; sisters, brothers, aunts, and uncles.

Lastly, I would like dedicate this dissertation to the purest love in my life; *my mother, my wife, and my son.*

1.0 INTRODUCTION

This chapter explains the motivation for the research topic studied in this dissertation, the main objectives, and provides an overview of the document organization.

1.1 DIFFERENTIATED CLASSES OF SERVICE

Over the last decade, many services have migrated to be automated and delivered over telecommunications networks as it becomes more convenient, *e.g.*, e-commerce, e-government, banking, tele-education, etc. In the near future, more services are expected to be carried over telecommunications networks as well (*e.g.*, smart grid). The migration of these services has been facilitated by the synergy of IP and optical technology. The nature of the Internet Protocol (IP), being flexible and adaptable to accommodate new applications (*e.g.*, overlays), has been one of the appealing factors to new applications to be built on top of the IP layer. In addition, the huge capacity offered by optical technology to transport high speed IP traffic was the key to carry the growing demand and reduce transport cost. Also, the advancement of control and traffic engineering technologies such as Multi-Protocol Label Switching (MPLS) and Generalized-MPLS (GMPLS) has complemented the layered architecture to a great extent. Beyond these features, however, there might be some hurdles related to service-specific requirements which are not supported by default.

Over time, the role telecommunication networks play in our lives has been increasing significantly. Services provided over communication networks have become an indispensable necessity and part of everyday activity for individuals, industries, businesses, and governments. Thus communication networks come to be an important and critical infrastructure

for overall social welfare and economy [1]. This role necessitates networks to be continuously available to use, reliable, and immune to disruption and failures. However, a perfectly resilient network is unachievable.

One of the major threats to communication networks are unintentional failures. As reported, unintentional random failures in communication networks are part of everyday actions in large carrier networks. Generally, failures in fiber-optics networks are more severe. A single fiber cable/link bundles 160 wavelengths, each with capacity of 10, 40, or 100 Gbps. Hence, a single fiber link failure leads to a tremendous amount of traffic loss. Thus, networks have to be supported by automated fault tolerance capabilities to maintain services in the presence of failures. In recognition of this, a considerable amount of research effort has focused on the concept of resilience in telecommunications networks. This has been encouraged by the significant role these networks play whereby network disruptions undermine societies capabilities and may lead to societal harm. On the other hand, it is encouraged by network operators who aim to maintain profitable operation in a cost efficient manner.

In general, modern telecommunications networks are accommodating various types of services. These services typically have different requirements in terms of Quality of Resilience (QoR) as well as Quality of Service (QoS), and security. QoR describes some parameters related to how a service maintains connectivity such as service availability, recovery time, percentage of recoverable traffic, etc [2]. These requirements are commonly specified in the service-level agreement (SLA) between the customer and the network provider (the carrier). Availability is a common metric of resilience [3]. It indicates how often a network or a service is in an operating state. Network or service availability is substantially influenced by the implemented recovery scheme. When a failure occurs, the recovery scheme defines how the affected traffic is rerouted in the unaffected part of the network. Redundant resources along the failure-free path must be available to use. The more redundancy added to the network, the higher the availability can be achieved. However, minimizing this redundancy would increase the likelihood to accommodate more traffic. Thus, on one hand, the providers seeks to optimize the utilization of network resources. On the other hand, it is good for the provider to satisfy customer requirements as agreed upon and avoid compensation for violations of the SLA reliability terms. More frequent violations may not only cost the provider the agreed

upon penalties, but it may affect market reputation.

Commonly, availability is expressed by the number of nines (*e.g.*, 0.9 one nine, 0.999 three nines). It also can be translated into the expected downtime (DT) per a specified period (*e.g.*, 36.5 days/yr, 8 hrs/yr, respectively).

The traditional approach to improving availability in systems is to add parallel redundancy [4], which in the context of typical optical backbone networks would imply implanting additional links and possibly nodes to the network topology to support additional parallel routes. However, adding links to nationwide or continent wide backbone networks simply to improve availability is difficult to economically justify. Furthermore, only a small number of users and services need very high levels of availability and these users/services produce only a small fraction of the total network traffic.

Currently, the majority of customers are satisfied with moderate availability levels (*i.e.*, around two 9's). Other specialized services may require higher QoR levels. Among others, critical service communications have more stringent requirements (*i.e.*, four to six 9's). This can be necessary for national security or public safety purposes (*e.g.*, emergency calls, networked medical systems [5], smart grid[6]) or to prevent severe financial losses (*e.g.*, banking, expedite mail services) [7, 8]. These services require service availability in the range of four to six 9's, which might not be achieved using basic protection schemes *e.g.*, 1+1 [9].

In general, the network has to be fully capable of providing differentiated services and properly provisioning them in a cost efficient manner, and taking into account these requirements may overburden the design process. The basic idea, instead of dealing with a single service, a network provider may offer multiple classes of QoR, each with different level of availability *e.g.*, [Gold, Silver, Bronze]. Essentially, the current approach is to take the physical network availability as a given and deploy redundancy and restoration techniques at various layers to provide QoR classes with different fault recovery capabilities and availabilities.

In literature, a considerable amount of research has studied service differentiation in communications network design [10]. Most of these works focus on assigning different protection and restoration schemes to flows based on their QoR classes at a certain layer [11]. For example, the gold class traffic is assigned a dedicated backup path and silver class gets

a shared backup path restoration, while the bronze class is not protected. A study from Verbrugge et al. [12] estimates the availability of each class, averaged over all node-pairs paths, for the 28 nodes/41 links European network (EPAN) to be: 0.99894 for gold, 0.99731 for silver, and 0.97746 for bronze class. Note that, other combinations of redundancy and recovery configurations are also possible and may provide different availability values.

In general, there are some limitations of the existing approach. First, the range and the spacing between availability classes for existing approaches are somewhat narrow. Both need to be extended to cover a wider range of classes. This extension should also account for overall cost of the design and avoid unneeded redundancy. Second, the recovery scheme of the gold class is insufficient to support extremely high availability levels (*e.g.*, five or six 9's). A higher configuration of dedicated protection (*i.e.*, 2:1 or 3:1) is needed. Also, reserving adequate sharable spare capacity to restore traffic from multiple simultaneous failures might be an alternative option. Both approaches, however, lead to inefficient utilization of network resources. The third limitation is related to the application of these approaches to layered networks. With just few exceptions, most of the existing approaches suffer from the crosslayer mapping issues discussed in the literature as without full knowledge of the physical layer and the mappings between layers no hard guarantees on availability can be provided (*i.e.*, due to fault propagation) [13].

1.2 MOTIVATION

Although the problem of providing quality of resilience classes has been investigated, the emergence of new services with different resilience requirements calls for a reconsideration of the problem. A research effort is needed to consolidate existing approaches with other techniques in order to satisfy stringent availability requirements and widen the range of availability values in layered networks in an economical manner.

1.3 OBJECTIVE

The objective of this research is to study the impact of heterogeneous link availability on services end-to-end availability in multilayer networks, and develop models that take advantage of this notion and provide different levels of availability to services of different classes of resilience. Our focus is on multilayer wide-area transport networks.

1.4 ORGANIZATION

The remainder of this dissertation is organized as follows: Chapter 2 provides a brief background on multilayer networks and availability metrics followed by a review of the related availability-based network design and crosslayer mapping literature in Chapter 3. Chapter 4 presents the thesis statement and the research contribution. Chapter 5 introduces our approach to the problem of how to provide high levels of availability in an efficient manner, namely the spine concept. It also presents our proposed spine link selection design problem. Chapter 6 extends the spine concept to layered networks. In this chapter, we propose our joint routing-mapping design problem. Finally, Chapter 7 concludes this dissertation.

2.0 BACKGROUND

This chapter provides a brief introduction to multilayer networks. It also introduces the concept of survivable mapping, the Quality of Resilience (QoR) classes, and some preliminaries on the availability metric and the nature of failures in communications networks.

2.1 MULTILAYER NETWORKS

Telecommunication networks are commonly perceived as multilayer networks, in which each layer has different capabilities and performs different functionalities [14]. Among others, IP-over-WDM networks have become the most prevailing architecture in today's carrier networks. In this dissertation, we consider a multilayer architecture with connection-oriented layers such as the popular IP/MPLS-over-WDM architecture.

2.1.1 IP and WDM Technology

WDM technology makes use of optical fibers exceedingly effectively. It exploits the capacity of optical fibers more efficiently by allowing multiple optical channels, modulating different wavelength carriers, to be combined (multiplexed) and carried on the same fiber simultaneously, in which a fiber becomes a multi-wavelength transmission medium. It successfully provides the large bandwidth needed to satisfy the growing demand. As a consequence, WDM-based networks have become the dominant technology for long-haul transport networks. The main elements in optical networks are optical line terminals (OLTs), optical add/drop multiplexers (OADMs), optical crossconnects (OXCs), and optical line amplifiers

(OLA). An OLT multiplexes/demultiplexes wavelengths into and from an optical signal. An OADM is responsible for selectively dropping incoming traffic and adding outgoing traffic from and to the optical signal. It switches traffic between metro and long-haul networks. It adds traffic from a local metro network to a specific wavelength/s in the long-haul network and drops traffic destined to the local network also from a specific wavelength/s. Also, it enables long-haul traffic to bypass local traffic. A reconfigurable OADM (ROADM) can be utilized to enable dynamic add and drop multiplexing to and from the long-haul network (*i.e.*, select the desired wavelength to add to and drop from). This allows flexible and independent planning of wavelengths in each network. OXC performs switching functionality within long-haul networks [15]. Typically, a lightpath between a node pair occupies the same wavelength in each fiber along its path. This is known as the wavelength continuity constraint. Careful planning of the available wavelengths is essential to ensure efficient utilization of wavelengths. Alternatively, the continuity constraint can be relaxed if the OXC is equipped with a wavelength converter that switches the wavelength of a lightpath to another free wavelength in the next fiber. This reduces the complexity of the routing problem and improves network utilization.

Internet Protocol (IP) is the most widely used networking protocol and can be deployed over a variety of networks (*e.g.*, SDH, ATM, WDM). IP networks transport data in packets between two node ends based on the destination address stored in a packet header. For an incoming packet, an IP router checks the destination address in packets headers and then forwards incoming packets onto outgoing links based on the corresponding next-hop entries stored on its routing table. An IP network is a packet-switched network. It is a connectionless protocol, offering a best-effort service with no guarantees for end-to-end QoS and QoR. Typically, IP routing follows weighted shortest paths.

Today, Multiprotocol label switching (MPLS) technology is commonly used along with IP networks to provide connection-orientation and a means to support QoS [16]. MPLS provides a label-switched path (LSP) between nodes. Packets are given labels associated with a LSP. An MPLS-enabled router, known as a label-switched router (LSR), maintains a label-forwarding table. Upon receiving a packet, an LSR reads its label, and based on the label entry in the forwarding table, the LSR replaces the label with the outgoing label

and forwards the packet to the appropriate outgoing link. A stream of MPLS packets can be grouped into a forward equivalence class (FEC). It is also possible to use multiple FECs between the same node-pair, and each FEC can follow a different LSP. Hence packets with different service class (*i.e.*, QoS or QoR) can be distinguished and routed differently. This also gives a means to perform traffic engineering [15].

2.1.2 Crosslayer Mapping

An IP/MPLS-over-WDM network is composed of two layers; the bottom layer is the optical network and the upper layer is the IP network. They are also known as the physical and logical layers, respectively. Each layer has its own sets of nodes and links. In the optical layer, optical fibers interconnect optical nodes, which can be equipped with optical cross-connect (OXC) to perform switching functionalities. A connection between two end nodes (*i.e.*, a *lightpath*) may traverse one or more optical fibers and intermediate routers and may occupy one or more channels (wavelengths) on each fiber. The IP topology is embedded onto the optical network. Each IP node (router) is attached to an underlay optical node. A logical link between two logical nodes is realized by a an underlay lightpath between the corresponding node-pair.

The problem of finding an eligible lightpath to implement a logical link is called *cross-layer mapping a.k.a, crosslayer routing*. The utmost design objective is to attain optimal utilization of network resources including link capacities and switching capabilities [17, 18]. Inappropriate mapping may lead to create loops in which some physical links are used more than once by the same logical link or path. This scenario leads to an inefficient use of resources. An example of this case is shown in Figure 2.1, where an upper layer flow path (in blue) is routed through two logical links that share a common physical link (in dashed red) in the lower layer, and hence inefficient redundant resources are reserved on this link. This problem is known as backhaul mapping. A completely free-loop mapping depends to a great extent on the upper layer topology [13]. In addition to utilization efficiency, survivability is a big concern in crosslayer mapping. A survivable mapping is the one that ensures logical network connectivity after any single (or multiple) physical layer failure [19, 20]. Figure 2.2

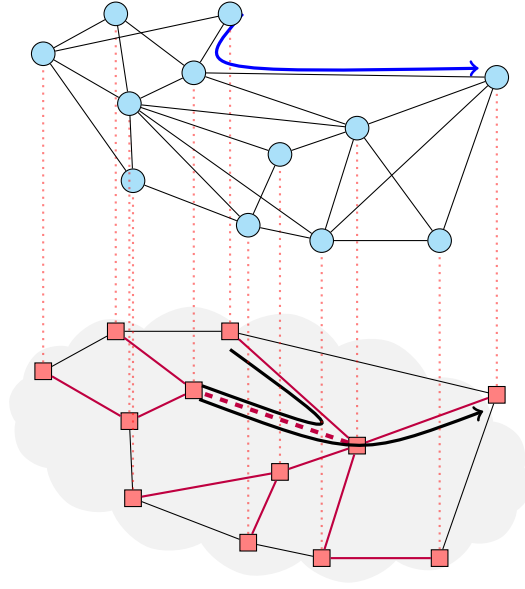


Figure 2.1: An example of backhaul mapping.

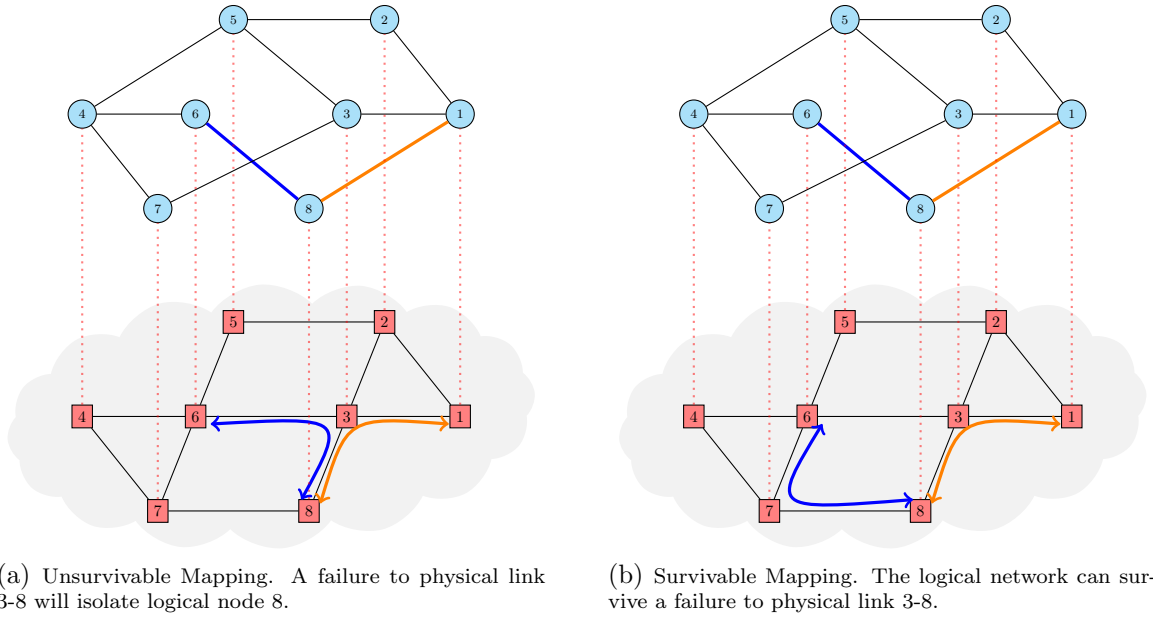


Figure 2.2: An example of survivable mapping.

shows an example of survivable and unsurvivable mapping with respect to logical links (1-8) and (6-8).

2.2 QUALITY OF RESILIENCE (QOR)

Resilience is a general terminology that describes network ability to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation. Resilience comprises survivability, fault tolerance, disruption tolerance, dependability, performability, and security [21]. Notwithstanding the wide range of resilience attributes, the concept of Quality of resilience (QoR), typically, has only been related to the service availability and its relevant measures: MTTR, MTBF, and traffic loss ratio [2]. There is a number of similar concepts in the literature incorporating both qualitative (*e.g.*, type of protection or restoration, covered failures, etc) and quantitative (*e.g.*, availability, recovery time, percent of restorable capacity, etc) survivability and resilience attributes. For example, quality of service and protection (QoSP) focuses on the recovery scheme and its quality (*e.g.*, recovery time, layer, etc.), then a single parameter is obtained to measure the QoSP. Another example is Differentiated reliability (DiR) that focuses on the number of recoverable failures with respect to each service. There is a great overlap between these concepts, but the main difference is the point of view in which options are chosen and evaluated. In here, our focus is on the availability metric and hence QoR.

2.2.1 Availability Model

Availability is one of the most common metrics related to resilience. Formally, the availability of a system (*e.g.*, service/component/path), as defined in [22], is “*the ability of a system/service/component to perform its required function at any instant of time within a given time interval*”. The actual availability of a component can be computed based on historical component failure data as the ratio of the aggregate up time to the total time the

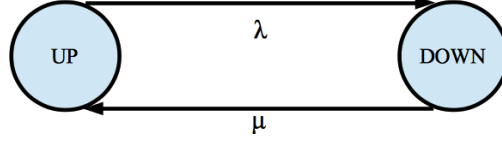


Figure 2.3: The Markov state model for availability.

component was in service,

$$A = \text{uptime} / (\text{uptime} + \text{downtime}) \quad (2.1)$$

Nevertheless, we need to estimate the availability level promised to be provided in the future. For this, one can use the instantaneous availability, $A(t)$, which is defined as “*the probability that a system is in an up state at a given instant of time t* ”. Further, we may be interested in the availability in a long time range (defined over the service lifetime) or the steady-state availability (*a.k.a.*, asymptotic availability) which is defined as the probability that the system is found in an operating state at any time in the long run future.

The most common model for availability is the Markov state model. In general, there is a number of assumptions that are commonly considered here: (1) a component can be in one of two states: either up or down status, (2) components fail independently (with the exception of known SRLGs), (3) Failures arrival and departure are independent processes, and each process has an exponential distribution with a constant mean. Thus failures in an availability model can be considered as a Poisson process. To illustrate this, consider the Markov model for a single component availability shown in Figure 2.3 [23]. It includes two states: *UP* and *DOWN*. The system moves from *UP* to *DOWN* state with a transition rate λ which is the failure rate, and transitions back to *UP* state with a repair time rate μ . Let P_{UP} and P_{DOWN} be the probabilities that the system is in *UP* and *DOWN* state, respectively. Then, by solving the equilibrium state probability of the *UP* state, we can compute the system availability as:

$$A = \lim_{t \rightarrow \infty} A(t) = \frac{\mu}{\lambda + \mu} = \frac{MTTF}{MTTF + MTTR} \quad (2.2)$$

where $MTTR$ is the mean time to repair ($= 1/\mu$) and $MTTF$ is the mean time to failure ($= 1/\lambda$). Given that $MTTF = MTBF - MTTR$, where $MTBF$ is the mean time between failures, we can rewrite (2.2) as

$$A = \frac{MTBF - MTTR}{MTBF} \quad (2.3)$$

The term unavailability is the probabilistic counterpart of the availability:

$$U = 1 - A \quad (2.4)$$

Commonly, availability value is expressed in the number of nines (*e.g.*, 0.9 one nine, 0.999 three nines). It also can be related to the expected downtime (DT) per a specified period (*e.g.*, 36.5 days/yr, 8 hrs/yr), which is sometimes a more convenient measure.

$$DT_{min/yr} = U \times 365_{days} \times 24_{hours} \times 60_{min} \quad (2.5)$$

To compute system steady state availability and unavailability of a system composed of n components in series, simply we can use reliability block diagram (RBD) theory as,

$$A_{series} = \prod_{i=1}^n A_i \quad U_{series} = 1 - \prod_{i=1}^n (1 - U_i) \approx \sum_{i=1}^n U_i \quad (2.6)$$

where A_i (U_i) is the availability (unavailability) of the i th component. For a system composed of n components in parallel,

$$U_{parallel} = \prod_{i=1}^n U_i \quad A_{parallel} = 1 - \prod_{i=1}^n (1 - A_i) \quad (2.7)$$

Now consider a communication path between two nodes as the evaluated system, and it comprises a number of links/nodes or hops as the components. Also assume that this path is a working path (WP) and it is protected by a dedicated backup path (BP). Based on these equations, we can state some general properties of availability measure in series-parallel system:

- Path availability is inversely proportional to its length and the number of hops [24].
- Path availability of a system connected in series is strictly upper bounded by its lowest component availability,

- End-to-end path availability of a disjoint WP and BP pair connected in parallel is strictly higher than the availability of the largest path availability.

Lastly, it is worth mentioning that availability and reliability are mistakenly considered interchangeable, however it is important to distinguish them because they have different meanings. Reliability is defined as “*the probability that a component will maintain its function over a specified period of time τ* ”. More precisely, it is the probability that no failure occurs in the period $[0, \tau]$, *i.e.*, $R(\tau) = Pr\{\text{no failure in } [0, \tau]\}$. Mathematically, it can be expressed in terms of the failure density function $f(t)$ as in (2.8). So by definition, reliability only considers how frequent failures come to the system and ignores the repair process. Also, the expectation of $f(t)$ gives also the mean time to failure ($MTTF$), shown in (2.9).

$$R(\tau) = 1 - \int_0^{\tau} f(t)dt \quad (2.8) \qquad MTTF = E[f(t)] = \int_0^{\infty} t \cdot f(t)dt \quad (2.9)$$

It is worth mentioning that assuming failure arrival times and repair times to be exponentially distributed simplifies the mathematical expression to a great extent as both processes will have constant means. Otherwise, availability evaluation for complex networks and its use in network design might be intractable. The exponential assumption is widely accepted by the research society even though failure and repair processes were found in some cases to follow different sup-exponential distributions (*e.g.*, Weibull distribution [25, 26]). The impact of having only the repair times non-exponentially distributed is negligible as $MTTR \ll MTTF$, which is expected to be the case. However, we have no clear understanding of the impact of having different distributions of these processes on estimating availability [25]. In addition, despite the wide use of this availability metric, it only provides the expected aggregated downtime over a course of time. It does not indicate the number of times an element will be in a failure state [3, 27], or how long it continues to work before failure (*i.e.*, continuity [28]). Furthermore, the accuracy of the evaluated service availability depends on the time period upon which availability is calculated. The longer the period, the more accurate the availability.

2.2.2 Classes of QoR

The concept of offering differentiated levels of products or services, in which these levels would have different costs, is a well recognized concept in the industry. The cost here denotes a two-sided matter; the price paid by the customer and the expenses incurred by the provider. The pricing for each level of service should commensurate the expenses in order to make offering such differentiated services profitable. Typically, a customer may choose a level of service that suits its needs and be willing to compensate the provider (carrier) for the specified quality provisioned. The differentiated cost is actually what provides the legitimate basis for service discrimination from the legal perspective as well as the incentive for the provider to offer such a service. There are many examples from industry that follow this concept. For example, a telecommunications network that accommodates a wide range of heterogeneous services and applications, each comes with different requirements in terms of capacity/bandwidth, minimum availability level, delay, etc, supports different levels of quality of service (QoS) [29]. A more analogous example would be the support of reliability classes in electrical power supply [30]. Here we consider service differentiation in communications networks where connections or flows between the same source-destination might have different requirements in terms of availability, and thus were classified in different QoR classes [31]. Typical availability requirements range from 99% to 99.9999%. The required availability level by a service depends on the service tolerance to outages and how much the customer is willing to pay for the connection. Typically, the required service availability, along with other QoS parameters, is explicitly stipulated in the SLA between the customer and the provider. It is up to both parties to negotiate this cost and associated penalties. In addition, it includes the strategy for monitoring the compliance of the agreed upon parameters. Monitoring can be done through direct access to customer (*e.g.*, web portal) or through a trusted third party to resolve any possible dispute [32]. The SLA should incentivize the provider to satisfy customer requirements as agreed upon and avoid compensation for violations of the SLA terms. More frequent violations may not only cost the provider the agreed upon penalties, but it may affect market reputation.

2.2.3 Failures in Communication Networks

Outages in communication networks, as well as in other large-scale systems, are common. These outages can be planned or unplanned. Planned outages can be scheduled in advanced for carrying out routine maintenance. Typically, equipments are shut down and service is provided over an alternate path or network, or completely interrupted for a predetermined period of time. On the contrary, unplanned outages occur randomly and without any prior knowledge of occurrence time and duration. They are commonly known as random failures. A random failure can be caused by a network element defect, malfunctioning, sabotage, fire, etc. Such a failure may disrupt communication services. Unfortunately, despite great efforts from network operators to maintain operated and physically protected networks, failures are inevitable and may lead to severe consequences.

All failures are not the same. They vary in the cause and the severity of the consequences. Some failures are persistence *e.g.*, a cable cut, a power supply failure, etc, and may need physical repair/replacement. The severeness of such failures depend on the time needed to repair (TTR). Other failures are soft, transient, and have minor impact, where a failed element can be recovered within short period of time *e.g.*, reboot a router or a line card [7].

In high speed transport networks, a failure in a network node may lead to huge loss of traffic and undermine network connectivity. A failure may affect the node partially *e.g.*, a failed line card, or completely isolate the node causing all originated, destined, and transited traffic to be dropped. Fortunately, node failures are not frequent. According to Bellcore report (1991) [33], failure rates for optical transmitters and receivers are 10867 FIT and 4311 FIT, respectively, where FIT denotes number of failures in 10^9 hrs. This means that an optical node is roughly susceptible to 0.133 failures/year. On the other hand, fiber optic cables have relatively lesser capacity but experience more frequent failures occurrence than optical or electronic nodes. A 1000 miles cable with a cut rate of 501142 FIT/1000 sheath-miles is susceptible to 4.39 cuts/year. A network like Level-3 [34] with 57,000 of long-haul cable miles is susceptible to a failure every 1.5 days. In addition, typical node equipment MTTR is ~ 2 hrs while the average for a cable cut is ~ 12 hrs. Hence the aggregate loss due to cable cuts would become more significant over time. To get more insight on what cable

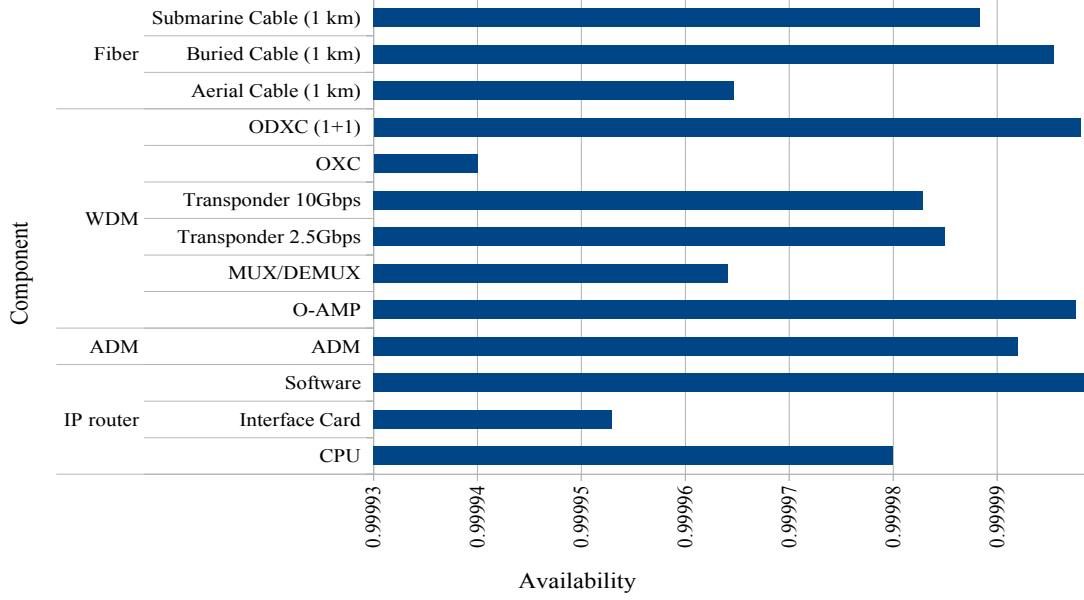


Figure 2.4: Typical Components Availability.

availability really weighs, Figure 2.4 provides estimates of equipments availability values based on experts from industry, as reported by Verbrugge et al. [12]. Note that cables availability shown in this figure is for 1 km. A 1000 km aerial cable availability goes down to 0.96471. Thus cable availability depends heavily on its length as well as its type.

It is widely agreed that cable damage is the dominant and the most disturbing type of failure in communication networks. NRCS annual report [35] shows that 43% of the total outages (reported to FCC) over a 12 years period were due to failure of different types of cables. The main reason that accounts for 57% of reported cable damage was cable dig-up during construction work. In addition, Snow [36] shows that 28.6% of the reported failures in telephone networks were caused by a cable cut excluding other cable failures causes. Other reasons include defected cable connector or amplifiers, rodents, natural hazards, car accident, or vandalism [34, 35, 37]. In addition, Snow found that cable cuts are more frequent in some states of the US than others, which can be attributed to the differences in weather and the frequency of construction work. Thus cable availability depends also on the terrain it traverses in which it is likely to impact both the frequency of failure occurrence and the

repair process. Crawford [38], in his study, found that the average MTTR for 160 cable cuts reported was 14 hours but with a high variance in which some repair cases reached to 100 hrs. Snow [36] found that high population areas might have shorter average MTTR and this was attributed to the proximity of repair parts and teams.

It is also important to distinguish between failures across layers. Optical network failures are knowingly considered more severe than failures affecting the electronic layer as the optical layer has larger capacity and processing capability. For example, a single logical link along its underlay path would be mapped to one or more wavelengths on each fiber it traverses. The traffic loss of a single logical link is much less capacity than a loss of a single fiber optic cable with capacity in the order of terabits per second. Also a failed optical cable may cause more than one logical link to fail simultaneously [39].

One of the major obstacles regarding research on this topic is the lack of publicly available data due to proprietary concerns [5]. As a matter of fact, most of MTBF, MTTR, and availability values used in the current literature are either two decades old (*e.g.*, Crawford’s reported 1993 data [38], Snow’s analyzed telephone networks reports collected between 1992-1996, To and Neusy [33] reported metro ring networks failures data collected in 1991), collected over small research or educational networks (*e.g.*, UNINETT [26]), collected from an IP layer network (*e.g.*, a 2002 dataset [39]), estimated (*e.g.*, [12]), or based on partially reported incidents [36]. Thus, there are many limitations in these datasets.

2.2.4 Techniques to Improve Equipment Availability

As a first step to improve network and services availability values, one should work on improving availability at the component level. Afterwards, the second step is about how to provide the services using network resources and recovery mechanisms. Recall that over a single path the overall service availability is upper bounded by the minimum component availability along the service path. Hence network operator should work on improving components with lower availabilities. This necessitates careful evaluation of network components availability to allocate the maintenance budget in an effective way [40]. Component availability can be improved simply by reducing the repair time (*i.e.*, MTTR). This can be achieved

through efficient management of work force, locating spare parts in nearby locations, training to improve labor productivity in a timely manner [41, 42], and following best practices for maintenance [8, 43, 44]. Snow [36] reported that there is a correlation between average labor salary and failure durations indicating that a highly skilled maintenance team would require shorter repair times. In addition, he found a strong correlation between the proximity of maintenance centers and failures durations.

In addition, reducing the frequency of failures (*i.e.*, MTBF) can effectively boost the component availability. Techniques for this purpose include adding redundant components in parallel (*e.g.*, redundant power source, router interface cards, cooling fans, management software), replacement with a new high availability equipment as, typically, different models of the same component might have different fault characteristics, physical protection (*e.g.*, bury cable, add caution labels on trench) [41, 42, 45]. Overall, the choice of suitable option, to a great degree, is a cost dependent [8, 46, 47], yet there is evidence of a clear correlation between the number of failures and the total investment in the network [36].

For fiber links, there can be multiple options for improving link availability. Fiber cables can be laid in ducts, direct-buried underground, or mounted on overhead pole-lines. The choice of the best strategy is highly dependent on the terrain (metropolitan, rural, forest, etc), weather, proximity to highways, and the cost of installation. Link availability also may be affected by other factors, such as the depth of the trench for the buried cable or the height of the poles for aerial cables, isolation type: PVC or armored shielding, number of amplifiers along the cable, physical protection added to the site, and warning markers [37, 48, 49].

2.3 SURVIVABLE NETWORKS

Network or service availability is substantially influenced by the implemented recovery scheme, which is why availability is a measure of survivability. Formally, survivability can be defined as “*the system’s ability to continuously deliver services in compliance with the given requirements in the presence of failures and other undesired events*” [50]. Practically, survivability defines the recovery mechanisms used for protection and/or restoration of the network or

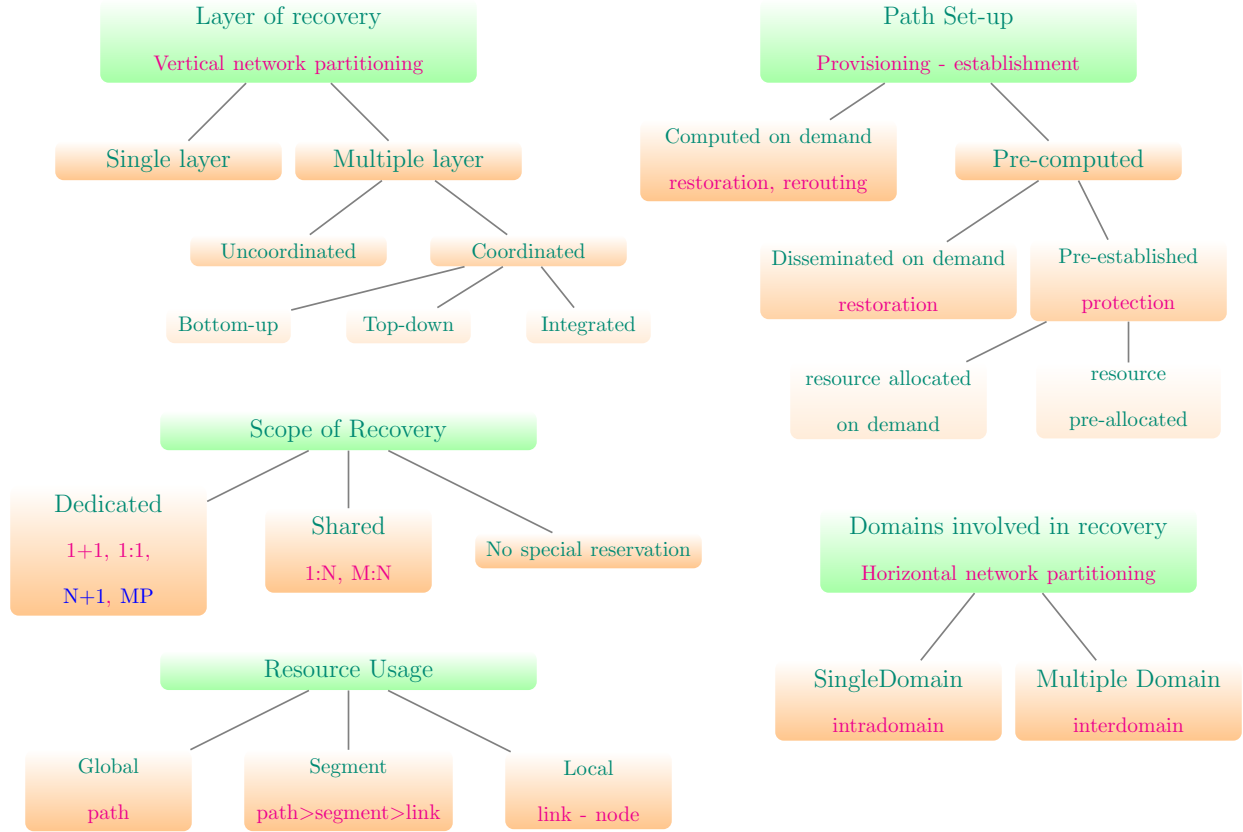


Figure 2.5: Classification of recovery mechanisms [51].

the service when failures occur. Recovery mechanisms describe how the ongoing traffic will be redirected/switched in failure scenarios.

2.3.1 Recovery Mechanisms

When a component along a working path (WP) fails, the affected traffic that uses this component will be rerouted through a backup path (BP). The detailed recovery procedures must consider some essential design criteria.

Figure 2.5 provides a broader classification of recovery mechanisms based on multiple criteria. Each criterion provides options that should be integrated with options from other criteria, *e.g.*, a scheme for *intra-domain* routing can be designed with *dedicated path protection* at the *physical* layer. The first criterion is the domains involved in the communication.

Domains are equivalent to autonomous systems. Each domain is owned and operated by a different provider. Based on this, a recovery mechanism can be designed for connections within a single domain (intra-domain) or multiple domains (inter-domain). The recovery methods of both scenarios might seem the same, however the multi-domain design is constrained by the owners' reluctance to share topology and routing information.

Second, recovery can be provided at a single layer or multiple layers as communications networks have been regarded as multilayer networks. Consider for example, IP/MPLS-over-WDM networks. Connections can be protected at the optical layer (WDM) with considerably faster recovery but with coarser granularity at the wavelength channel level. On the other hand, logical layer (IP/MPLS) offers finer traffic granularity at per-flow level but takes longer recovery time. Generally, recovery mechanisms are used at multiple layers in which each layer can protect itself and prevent failures to propagate to other layers. However, there are some scenarios where single layer recovery is insufficient and coordinated multilayer recovery strategies are needed [52].

Thirdly, based on the scope of the recovery, mechanisms can be classified into global path, segment, and local node or link recovery. In path-based scheme, connections or traffic affected by a failure are rerouted using alternate end-to-end paths, however, local recovery schemes are designed to avoid a faulty link/node only. Segment-based schemes offer an in-between solution where a section of the working path (a chain of links and nodes along the WP called segment) is protected by a backup segment.

Recovery mechanisms can be further classified based on how backup path is set-up into *protection* and *restoration* schemes, and these schemes can be used on path, segment, or link basis. In *protection* techniques, backup resources are pre-computed and reserved in advance. These resources can be dedicated or shared. Dedicated backup resources reserved for a connection are only used by this connection; either continuously (1+1) or in case of a failure affecting the working path (1:1). Shared backup resources (1:N) are shared between a number of connections, N . If a connection occupies this shared resources due to a failure on its working path, then all other sharing connections are blocked from using this backup resource when needed and left unprotected. This scheme requires the sharing connections to be part of different Shared-Risk-Link-Groups (SRLGs), *i.e.*, SRLG-disjoint.

In general, the total spare resources reserved in shared backup schemes are much less than those used in dedicated backup protection. However, it requires more signaling overhead. Even though shared-backup protection can achieve high resource efficiency compared to dedicated protection, this comes at the cost of lowering connections availability. A trade-off can be achieved using advanced sharing schemes like, $(M:N)$ where working paths/links of N connections are protected by M shared-backup resources, with $M < N$.

Typically, in protection recovery methods, reserved spare capacity is left unused during normal operation which limits the network utilization. Unlike protection techniques, *restoration* techniques require no pre-reserved spare resources. In case of failures, backup paths (either end-to-end path or link-based) are computed on-the-fly and resources are reserved, if they are available. Off course, higher utilization level can be achieved using restoration method but this comes at the risk of not finding alternate paths and hence leaving connections unsurvivable to failures. Also, restoration requires longer recovery time for path computation and control signaling. In addition, protection schemes are suitable only for connection-oriented technologies such as WDM at the optical layer or MPLS at the logical layer. They are not applicable to pure IP layer where the option of setting-up a predefined end-to-end path or path pair is typically not enabled for wide area networks due to security concerns [53]. Here, restoration recovery is attempted at the IP layer when a failure is detected [7, 51].

2.3.2 Survivable Network Design (SND)

The problem of routing and assigning recovery options to services in communication networks is known as the survivable network design (SND) problem which has been extensively studied in literature. A wide range of proposals offering different approaches to the problem, and availability-constrained design can be considered as a subproblem of the general SND problem [10, 54]. SND aims to design the network with automated fault tolerance capability with respect to some design objectives. Typical design objectives involved in the SND problem can be summarized as follows:

Capacity objectives

Capacity Efficiency : the goal is to optimize the utilization of network resources.

Guaranteed Capacity : the design should guarantee that every connection (flow) gets its required capacity (bandwidth).

Acceptance/Blocking rate : increase the chance of accepting new requests.

Resource Overbuild : (*a.k.a.*, Redundancy) defined as the ratio of protection capacity to working capacity. The goal is to minimize the resource overbuild *i.e.*, minimize the spare capacity used for protection.

Load Balancing : avoid creating congested links or nodes.

Recovery objectives

Differentiation : network ability to support multiple QoS classes.

Availability : the design should provide some guarantees over the end-to-end path availability provided to services.

Restorability : increase the possibility of recovering disrupted traffic in a timely manner.

Recovery Time : some services are very sensitive to recovery time. In mesh networks, it ranges from 50ms to several seconds. This basically depends on the recovery mechanism and on which layer the service is provided.

Complexity and Scalability

The design should perform in a timely manner and scale properly for large networks and large number of connections. Complexity is measured by algorithm execution time and the volume of signaling overhead.

2.3.3 Differentiated Classes of Survivability

Table 2.1 shows an example of multiple survivability classes and the corresponding technical and economic factors for each class [55]. The classes are ordered from highest to lowest priority. Although our focus here is on availability-based design, it is important to address the trade-offs between the different objectives.

Simply, differentiation can be achieved by managing to have different recovery classes, each is configured with a different survivability scheme. For example, the highest class, on the first row in the table, can be given $(N+1)$ dedicated protection with N backup paths.

Table 2.1: An example of recovery classes in [55].

recovery	survivability scheme	scalability	cost	recovery time	spare capacity	comments
guaranteed	Dedicated protection (N+1), $N \geq 2$	very complex	extremely expensive	fastest	ultra high	no preemptive traffic use spare capacity
	Dedicated protection (1+1)	complex	very expensive	fast	very high	
	Dedicated protection (1:1)	complex	very expensive	fast	high	preemptive traffic may use spare capacity
	Shared (M:N) protection	medium	medium expensive	medium	medium high	
	Shared (1:N) protection	medium	less expensive	medium	medium	
	Pre-planned Restoration	simple	regular	slow	none	requires post-failure resource reservation and signaling
not guaranteed	Restoration	simple	cheap	slowest	none	post-failure re-routing
NA	Unprotected	very simple	very cheap	none	NA	
NA	preemptable	simplest	cheapest	NA	NA	preempted if resources are needed by other classes

Hence traffic belongs to this class can survive up to N concurrent failures which means high and guaranteed restorability level and fast recovery. However, this comes at the expense of high resource usage for recovery (cost) that lowers the efficiency of resource utilization. A lower class can be protected by (1:1) dedicated protection or shared protection with traffic in this class can survive any single failure. This of course consumes less resources than the first class, and in general shared protection is more capacity efficient, but slower. Yet, in all protection schemes reserved backup resources are either left unused during normal operation or used to carry duplicate traffic. To reduce the redundancy level, one can utilize the unused resources to carry preemptible traffic, that is, the lowest class of traffic that can be dropped if the resources are needed by other higher classes traffic. The survivability of this class is actually worse than even unprotected class. These classes are shown at the last two rows in the table. Intermediate classes can be offered with restoration recovery with no guarantees and comparatively longer recovery time needed to find alternative routes. However, these classes require no redundancy and can offer a moderate, and potentially high, degree of

survivability that mainly depends on the utilization level of network resources. Overall, it is clear that high survivability options cost more resources and reduce the network ability to accommodate new incoming services.

3.0 LITERATURE REVIEW

In this chapter, we survey the literature on supporting differentiated availability levels in network design with a variety of design objectives.

3.1 ROUTING ALGORITHMS

This section introduces the core of the routing problem from a technical perspective. There are three main approaches used in the literature to route connections or flows between the end nodes: mathematical programming, alternate path, and weighted link routing. Each approach handles explicit availability requirements differently. Here, we discuss these approaches highlighting how availability is handled.

3.1.1 Optimization Problem Approach

Using mathematical programming, we can find optimal paths for a set of connections. The optimality here is with respect to some objective (*e.g.*, minimum resources). Path availability can also be included. Let us consider the following simple formulation for the problem of routing a set of connections in an optical network, shown in eqs. (3.1) to (3.6) [56].

Given a network $G = \{V, L\}$, where V and L are the node and links sets, and given a set of connections, C , that need to be routed on the network, the objective is route each connection so that it meets or exceeds its target availability A_{sd}^t . The problem is formulated as follows:

Notation:

V	is the set of nodes and L is the set of links.
mn	a physical link between node m and n .
sd	connection (<i>e.g.</i> , a lightpath) between source-destination.
a_{mn}	availability of link mn .
A_{sd}^t	target availability for connection sd .
C	set of connections to be routed (it can be flows/logical links).
W_{mn}	capacity of link mn (<i>i.e.</i> , number of wavelength channels).
X_{mn}^{sd}	binary variable = 1, if connection sd uses link mn for its <i>WP</i> .
Y_{mn}^{sd}	binary variable =1, if connection sd uses link mn for its <i>BP</i> .

Objective function:

$$\underset{X_{mn}^{sd}, Y_{mn}^{sd}}{\text{minimize}} \quad \sum_{sd \in C} \sum_{mn \in L} X_{mn}^{sd} + Y_{mn}^{sd} \quad (3.1)$$

subject to

Flow conservation constraints :

$$\sum_{n|kn \in L} X_{kn}^{sd} - \sum_{m|mk \in L} X_{mk}^{sd} = \begin{cases} 0 & \text{if } k \neq s, d \\ 1 & k = s \\ -1 & k = d \end{cases} \quad \forall k \in V, (s, d) \in C \quad (3.2)$$

$$\sum_{n|kn \in L} Y_{kn}^{sd} - \sum_{m|mk \in L} Y_{mk}^{sd} = \begin{cases} 0 & \text{if } k \neq s, d \\ 1 & k = s \\ -1 & k = d \end{cases} \quad \forall k \in V, (s, d) \in C \quad (3.3)$$

Disjointness constraint:

$$X_{mn}^{sd} + Y_{mn}^{sd} \leq 1 \quad \forall (s, d), (m, n) \in L \quad (3.4)$$

Link capacity constraint

$$\sum_{(s,d)} (X_{mn}^{sd} + Y_{mn}^{sd}) \leq W_{mn} \quad \forall (m, n) \in L \quad (3.5)$$

Availability constraint

$$1 - \left(1 - \prod_{mn|X_{mn}^{sd}=1} a_{mn} X_{mn}^{sd}\right) \times \left(1 - \prod_{mn|Y_{mn}^{sd}=1} a_{mn} Y_{mn}^{sd}\right) \geq A_{sd}^t \quad \forall (s, d) \quad (3.6)$$

This is a link-path routing problem that assigns each connection c an eligible (1+1) dedicated path-pair between source and destination node-pair (sd) with the same required capacity on both paths –one unit– and with a minimum availability level of A_{sd}^t . Routing connections between a node-pair is realized by the flow conservation constraints (3.2) and (3.3) for working and backup paths, respectively. Constraint (3.4) ensures no common links between the path-pair. The main objective, (3.1), is to minimize the total working

and backup capacity used for routing the connections with dedicated path protection (DPP) constrained by a minimum availability target for each connection (3.6). While this problem is designed to provide an optimal solution, it is known to be an NP-complete problem. Besides that, it is clear that the availability constraints introduce non-linearity to the problem. This is only for the simplest formulation for routing with DPP path-pair. Further considerations of shared path protection (SPP) and/or hybrid method only increases the complexity of the problem. As a result, several proposed algorithms try to either adopt iterative-based heuristics to solve the availability-constrained routing problem or define general rules that dictate network availability and indirectly solve the problem. In the latter form, availability is used as an evaluation metric.

3.1.2 Alternate Path Method

The basic idea of this approach is to assign a path or path-pair to a connection chosen from a set of predefined candidate paths. To generate this set, a modified version of classical routing algorithms (*e.g.*, *Dijkstra*, *k*-disjoint shortest paths) can be used. Thus, finding a suitable path or path pair out of this set is achieved by enumerating through these paths. Then the path or the pair that provides the highest availability, or meets the target availability, is selected. However, it is also favorable to do this in a resource efficient manner.

For illustration, consider the algorithm introduced here as an example. The algorithm is based on fixed-alternate routing [57], in which for each node-pair, a number of M candidate paths (or link-disjoint path pairs) is precomputed. Candidate paths can be, for example, any combination of the options [56]:

- Shortest Path (SP): the path/s with minimum hop count, or Weighted SP with minimum total weight.
- Shortest Path-pair: path-pair with minimum hop count (min-min and/or min-sum), or with minimum weight.
- The Most Reliable Path (MRP) (or disjoint path-pair): path with minimum cost, $MRP = \{Path | \min \sum_{l \in path} C_l\}$, where $C_l = -\log a_l$ is the link weight.
- Others: this can include the second SP or MRP, third, and so on.

Note that the first two candidate path sets minimize the resource usage (*cost*) while the third case maximizes the availability. Additional customized routes can be used as well [58, 59]. The last set of paths may further introduce tradeoffs between the cost, availability, and load balancing. Thereby, each connection should be assigned a path (or a path-pair) that satisfies its availability target and uses minimum resources.

To ensure both objectives are maintained, Tornatore et al. [60] propose a scheme to assign these paths to connections arriving dynamically to the network. Starting from the path with the minimum resources SP, then try the MRP, a shortest path pair, a path-pair with one shortest path SP and one MRP, and then a path-pair with two MRPs in this specific order until the availability requirement is satisfied. The scheme also prioritizes backup paths that can be potentially shared between multiple connections, if a connection availability target is surpassed. First, one enumerates through all possible combinations of paths and identifies a subset of paths and/or path-pairs that satisfy the connection availability requirements. Then, enumerates through all paths in this subset and finds the path or path-pair that uses the minimum resources. Repeat this for all connections.

A different way to assess the set of candidate paths is to assign a proper cost for each path, such as $cost_p = (1 - \alpha)h_p + \alpha A_p$, where h_p is the path length, and α is a parameter between (0,1) [61, 62]. Know that, if we found a set of candidate paths and computed their availability values in advance, it is possible to use an ILP to select optimal paths with respect to some cost metric. This would change the NLP shown above to a path-path routing problem and the non-linearity would be avoided [63].

3.1.3 Weighted Routing Approach

Instead of generating candidate paths, we can route each connection using a weighted shortest path. By assigning a proper weight to each link, we can iteratively find suitable path or path-pair for each connection. The link weight can be any combination of link availability (*i.e.*, $C_l = -\log a_l$), link capacity, residual capacity of the link, and/or number of times the link is used. The routing can be applied twice or more; first to compute WP, then to compute a partial or fully disjoint BP [64–66]. If the availability target of a connection is not met using

a single path, then a dedicated path-pair is given, and typically it is required to be a fully link-disjoint pair (using $C_{l \in WP} = \infty$). However, if the assigned DPP yields an end-to-end availability greater than the targeted availability, a possible approach to reduce the amount of the reserved resource is to reassign a partially-disjoint path pair or a shared path protection. On the other hand, if the fully link-disjoint path-pair does not achieve the availability target of the connection, an additional protection path may be granted. The sub-problem is known as k -disjoint paths [67].

Another important issue in SPP is that in general the WPs of the sharing connections should be SRLG-disjoint to avoid an unrecoverable scenario [62, 67–69]. For this, one can use an SRLG identifier associated with each link. In addition, Miyamura et al. [70] propose a scheme for path selection for traffic in GMPLS networks based on varying the disjointness degree of links’ SRLGs. At this layer, physical network and cross-layer routing information must be attainable. SRLGs in this scenario involves both links that are physically adjacent (*e.g.*, use the same conduit) and logical links mapped to the same underlying link.

Know that, as mentioned earlier, path availability is inversely proportional to the path length and the number of hops. This indicates that in single path routing, shortest path is always the optimal path in terms of both availability and cost. However, this can be true for availability only if all links availability values are identical (or have very small differences), which is not always true. For example, Markopoulou et al. [39] show that failures at the IP layer in only 2.5% of the links account for more than 50% of the IP layer link failures. In addition, relying only on shortest path might lead to imbalance load distribution and link congestion.

3.2 AVAILABILITY-AWARE NETWORK DESIGN

3.2.1 Alternating Recovery Mechanisms

The most common approach to resilience differentiation is by assigning different recovery schemes to the different traffic classes in a fashion similar to Table 2.1, but with explicit

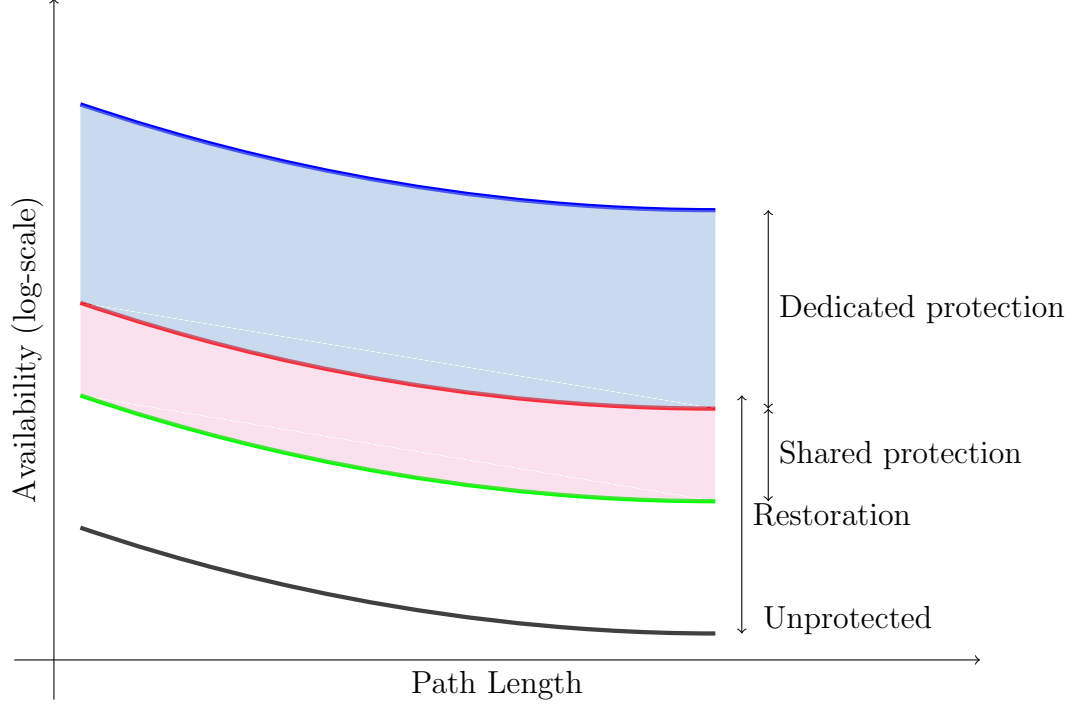


Figure 3.1: Availability comparison of basic recovery options [51].

availability constraints on the routing problem. Figure 3.1 gives a qualitative comparison between the basic recovery schemes availability. As we discussed in section 2.3.3, dedicated backup protection (DPP) can be used to provide high levels of availability. However, this comes at the cost of lowering network utilization and hence accommodating less connections. Also DPPs with more than one BP are deployed for connections with extremely high availability targets. This would further worsen network utilization. In order to reduce the cost of protection without violating the requirements, a number of options has been proposed for this purpose.

3.2.1.1 Shared Protection A direct method to assign shared path protection is to start from a feasible DPP-based routing that satisfies the requirements of all connections. Then, for each link find the set of connections whose BPs use this link and belong to different SLRGs (*i.e.*, their WPs are link-disjoint). After that, for each group of connections in this set, the DPP for these connections can be downgraded to shared protection. However, these

connections should be downgraded carefully without violating the availability requirement of any connection in the sharing group. Also, it is appropriate to start with the most utilized link [56, 58].

Another way to assign shared resources to connections follows this process: first, calculate a set of candidate path-pairs, compute their end-to-end availability, and remove path-pairs that do not fulfill the requirement. Then, enumerate through all eligible pairs and for each pair: 1) test if the links on the BP have enough (shareable) spare capacity, 2) availability target is not violated, 3) all other connections' WPs that will use the sharable spare capacity are SRLG-disjoint. If more than one pair satisfies these conditions, pick the one with lowest resources (*i.e.*, minimum hop count path-pair) [61, 62, 71, 72].

Alternatively, $M:N$ shared path protection scheme can offer a better trade-off between availability and resources. In this scheme, M BPs are protecting N WPs belonging to N different connections, with $M < N$. Thus, it can be used to increase connections availabilities with an increase in backup resources. Yet, the cost and availability values are still lower than DPP, which means that this scheme can expand the network differentiability by adjusting the cost-availability tradeoff [73]. For the routing problem, we first need to generate k -disjoint paths for each node pair. Then, follow the same procedures of assigning shared resources on each link as shown above with each connection given shared resources on multiple backup paths N . In general, the routing problem becomes more complex.

In the following some concepts related SPP are described briefly.

Link Shareability In the previous methods, it is important to note that the number of the connections in the sharing group controls the overall connection availability. If a connection, c , protected by SPP then its end-to-end availability is given by

$$A_c \approx 1 - (1 - A_{WP})(1 - A_{BP}A_{S_c}) \quad (3.7)$$

where $A_{S_c} = \prod_{l \in S_c} A_l$ and S_c is the set of links whose failure, along with c , causes a conflict on a link that is on the BP of connection c . Thus, we can adjust the sharing degree, $|S_c|$, on each link to control connections availability. This also can be done for each QoR class to support differentiation. For example, the gold class can be assigned a smaller sharing degree on all links than the silver class, hence connections belonging

to the former class can attain higher availability [66]. For the $M:N$ scheme, connection availability depends on both the number of sharing group M as well as the number of protection paths N .

Priority Priority-aware routing can be an efficient and inexpensive way to achieve differentiation. It is proposed to be used in resolving contention between connections competing for shared resources. Probabilistic availability models and priority-based routing schemes for $1:N$ and $M:N$ can be found in [74, 75]. It has been shown that this approach could slightly increase the differences in the availability levels between the sharing connections. In addition, it can be extended to restore affected traffic based on their priorities, *i.e.*, which connection to restore first.

Preemptable Traffic This concept introduces the class of preemptable traffic that may be evacuated (dropped) from its not-affected primary path to serve other higher priority connections whose WPs are down. In this case, we can further set availability levels apart from each other at no extra cost [76, 11, 77].

3.2.1.2 Restoration Restoration schemes require no reserved capacity for recovery. When a failure occurs, the network searches for eligible paths to reroute the affected traffic. The rerouting decision is subject to whether enough resources are available or not. Successful restoration, hence, depends merely on the utilization level of network resources. Restoration schemes are the most common recovery option in large carrier networks [78, 79] due to their low cost. However, successful recovery of disrupted connections is not guaranteed. If *full restoration* is not possible, *partial restoration* can be given in which only a fraction of the total demand capacity is restored.

Due to the nature of this scheme, accurate availability evaluation is difficult. There are some analytical availability expressions for restoration that appear in the literature, *e.g.*, [80]. Potentially, high availability levels (may exceed DPP) can be achieved because there is a chance to cover more high-order failure scenarios. However the overall availability levels, compared to other schemes, are not guaranteed and it depends in the utilization of network resources.

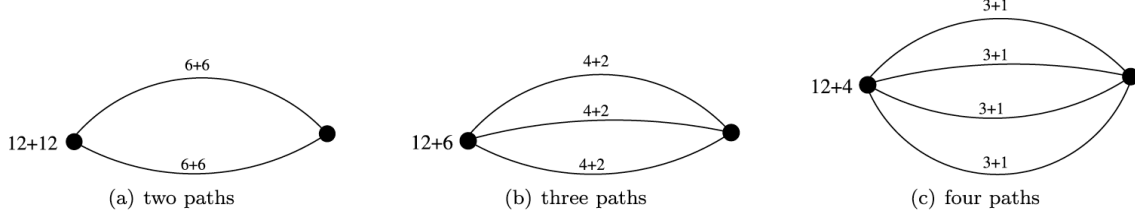


Figure 3.2: An example of self-protecting multipath routing.

3.2.1.3 Multipath Routing (MP) In an IP/MPLS network, it is possible to use two or more WP s to connect two nodes, in which traffic is split through these disjoint paths. Each path may be set up with dedicated spare capacity to provide protection in case one of the other WP s is down. Figure 3.2 shows an example of a multipath routing. It shows that the total spare capacity is reduced compared to the DPP, while the end-to-end availability is improved. Huang et al. [81] provides analytical models for evaluating connection and sub-connections availability. It shows that sub-connection availability may exceed DPP as more failure scenarios are recoverable.

Alternatively, a number of N primary paths and M backup paths can be given to a connection, with each WP carrying only a fraction of the connection traffic and each BP protecting a fraction of the traffic [82]. A single failure will affect only a small proportion of the total traffic, which can be easily restored using the M backup paths. By this diversity on WP s and BPs, the connection can be, at least partially if not full, recoverable against multiple simultaneous failures. Hence its availability can be improved with lower resource than 1+1 DPP. In addition, when a fraction of the bandwidth of the demand can be routed through another path, this reduces blocking rate caused by congested link/node and helps balancing the load in the network (assuming divisible capacity in WDM or configuring at MPLS). In general, by using this scheme, a high availability level is achieved while the overall network resource overbuild and blocking rate can be reduced [81, 82]. For the routing problem, many proposals use modified versions of the k -disjoint paths routing algorithm to find the eligible multipaths. Alternatively, Ma et al. [83] propose an ILP formulation for protection against dual-failures using the multipath mechanism.

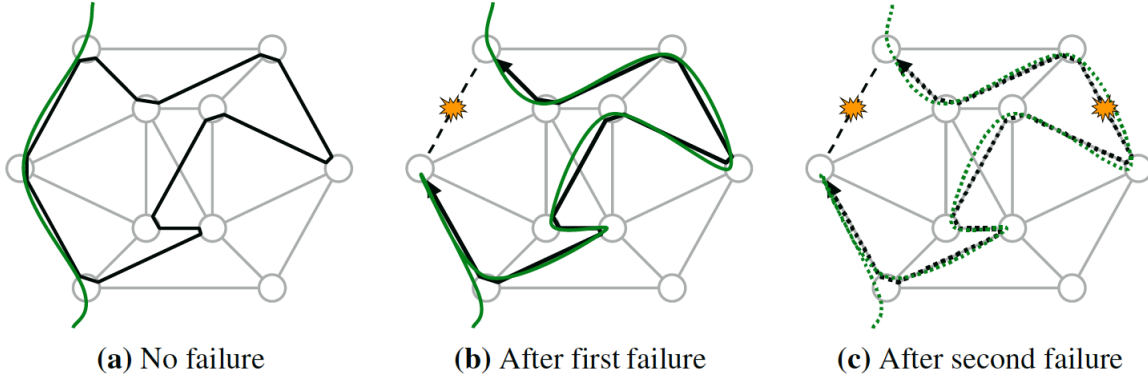


Figure 3.3: An example of a dual-failure scenario with p-cycle [84].

3.2.1.4 P-cycles Clouqueur and Grover [84] proposed P-cycles mainly to improve network restoration recovery time. The idea of P-Cycles is to pre-define a cycle on the network with sharable resources used only for protecting the connections between nodes that are on the cycle path. Hence emulating the ring protection in a meshed network. If one arc is down, protected traffic is routed through the counterpart arcs that complement the cycle (see Figure 3.3 as an example). Clouqueur and Grover [84] define a tradeoff between capacity efficiency and availability in p-cycles. On one hand, path availability is highly dependent on the size of the p-cycle; the smaller the cycle the larger the availability, and to a lesser degree on the number of the straddling links (*i.e.*, the links that are not on-the-cycle but their end-nodes are). On the other hand, limiting the size of the p-cycle reduces its capacity efficiency. The authors showed that paths traversing straddling links achieve higher availability (18-25% lower unavailability) than those on-the-cycle. And the availability of paths on straddling links depends on the size of the p-cycle. Accordingly, the higher class paths are routed exclusively on straddling links with enough protection capacity on the cycle, where it can be offered one or two on-cycle protection path. Paths of the lower class are routed either on-the-cycle or off-the-cycle, a route that can minimize the total spare capacity. Protecting both types of paths would definitely require more capacity on the p-cycle. Also a hybrid path (path that crosses on-the-cycle and straddling links) can be used with moderate availability [85]. Lastly, Sebbah and Jaumard [86] propose P-structures that extend the p-cycle concept

to an unrestricted structure, however no availability evaluation is given.

3.2.2 A Time Perspective Recovery

For any connection, the computed availability is not the actual availability that the connection yields at the end of its contracted period. Due to the probabilistic nature of the availability model, actual availability can be more or less than the availability evaluated prior to the service. Hence, there has been a research path that tries to exploit this marginal benefit between the designed availability and the actual experienced/received availability as the time proceeds. The aim of this approach is to reduce the redundancy (spare resources) while at the same time take advantage of controlling the actual downtime in the network by exploiting the knowledge of current network states, and hence differentiate connections at a finer time scale and potentially improve network utilization.

The basic idea is to reprovision each connection with a redefined target availability. Hence a new path or path-pair and an altered recovery scheme may be assigned to each connection accordingly. Reprovisioning can be triggered by the event of new connection/s arrival or departure [87–89], or triggered by the event of failure, exploiting the knowledge of failure events and experienced connections downtimes to better redefine new availability targets [88, 90, 61]. Alternatively, differentiation can also be achieved through a restoration scheme that restores connections in the event of failure based on their accumulated and maximum allowed downtimes [91, 92]. Furthermore, a hybrid scheme can be deployed to increase higher classes satisfaction rate while using less spare capacity compared to protection schemes [93, 94].

In general, the main drawback of this approach is the need to maintain the failure status for all flows and give real time routing decisions, which might not be scalable when dealing with large networks and thousands of demands [95].

3.2.3 Overall Network Availability

In this type of design, *network availability* is considered as the reference measure to resilience rather than individual connection availabilities. This probabilistic measure facilitates the

integration of a metric of interest into the design objective. The main idea is to define all mutually exclusive network states and compute the probability of being in each state.

Vajanapoom et al. [96] propose a survivable network design that aims to minimize the total *network risk*. The network risk is defined as the sum of the network state probability times the expected damage of this state. Each network state represents a unique failure scenario, and its probability is computed as the product of the unavailability of each failed component and the availability of all up components in the network state. Lee et al. [97] study the reliability of layered networks. The authors propose two algorithms to improve the reliability in layered networks; lightpath rerouting and logical topology augmentation. The two algorithms are based on the proposed reliability metric, *cross-layer failure probability*, that captures the number of cross-layer cuts due to one or more physical links failures and the probability of these failures. To improve the network reliability, the objective of the lightpath rerouting problem is to find the best way to reroute a lightpath, so that the *minimum cross-layer cut* is maximized and its probability is minimized. The logical topology augmentation problem aims at improving the reliability by adding a new logical link in order to reduce the number of minimum cross-layer cuts. Most recently, Zhang et al. [49] proposed an algorithm to maximize overall network availability by shielding a minimum set of links (*i.e.*, adding physical protection to improve link availability). In general, all these approaches do not support differentiated classes of resilience.

In the end, the current schemes try to trade cost for availability (or, in general, QoR) and hence by controlling this tradeoff, differentiation can be achieved. In addition, there is a complexity level associated with each scheme. This includes the time needed to provide a routing solution and the scalability of the algorithm used as well as the signaling required for setting-up, tearing down, and recovering the connections.

3.3 AVAILABILITY IN MULTILAYER NETWORKS

3.3.1 Mapping Link Availability

It is worth mentioning that the path selection problem with availability constraints in a multilayer network essentially starts from the physical layer, where the physical layer availability can be included in the crosslayer mapping (routing) problem. This problem is defined as the problem of finding an appropriate path for a logical link in the underlying physical network. Each link in the logical layer is mapped (routed) to a multi-link path in the underlying physical network. Then, logical link availability is evaluated as the availability of the corresponding physical path availability [98]. If protection is provided at the bottom layer, the logical link is mapped to two or more physical paths, and its overall availability can be treated as parallel configuration of multiple physical paths as in equation (2.7). Afterwards, computation of flow availability, the upper layer follows the same procedures. This method, however, accounts only for failures originating at the physical layer unless the availability of the components in the logical layer (*e.g.*, IP routers) are considered [79, 12]. An important concern here is related to the accuracy of the availability evaluation. This is because the mapping of the logical layer into physical layer might duplicate the usage of some links by a certain flow (*i.e.*, backhaul). This scenario leads to the inclusion of a link unavailability more than once in the availability calculation and hence underestimating the end-to-end availability [99].

Approaches for finding working and protection path are similar to those in a single layer *i.e.*, alternate path-based, or weighted routing [100, 101]. However, path selection requires careful mapping of overlay links/paths to avoid cross-layer correlated failures, in which multiple upper layer links/paths may fail simultaneously due to a single underlying failure *i.e.*, if diverse paths share a common underlying physical link [13]. Typically, logical networks are denser and, hence, it is more likely that multiple logical links will use a common single physical link (*i.e.*, these links will be in the same SRLG). Then, when this link fails, multiple upper layer links will fail simultaneously. Markopoulou et al. [39] show that, in the Sprint IP backbone network, 11.4% of failures occur at IP layer are correlated events where

multiple (at least 2 and at most 25 logical links) share the same physical link or router.

3.3.2 Types of crosslayer problems

Based on the entities who manage the layers, there are two types of routing problems. One that can have all topological information for both layers (*e.g.*, ISP owns the physical infrastructure), and one with limited information about the network layers (*e.g.*, a virtual network operator and a physical network provider). And generally, there are two approaches to provide redundancy in multilayer networks; 1) using traditional protection schemes at the physical layer. This approach maintains better use of resources [98]. 2) augmenting the logical network to enable recovery at the IP layer, and hence working on finer granularity which might improve the restorability [102]. To ensure high survivability, Zhou et al. [19] identify a set of spanning trees of the logical topology such that at least one of these trees remains connected after any physical link failure.

Once adequate redundancy is placed, recovery can be triggered at both layers, as shown in Figure 2.5, in a bottom-up, a top-down, or an integrated fashion. In the bottom-up approach, when a failure occurs, recovery is triggered at the physical layer and if it is unsuccessful, then upper-layer recovery is attempted. In the top-down approach, the optical layer recovery, which is typically faster, is delayed until attempted upper layer protection fails in recovering from the failure. The integrated approach chooses the best layer for recovery before any attempt. The advantage of multilayer protection is that recovery at both layers enables covering a wider set of failure scenarios that cannot be recovered using single layer protection. However, it might result in an inefficient use of redundancy where most of the failure scenarios will have duplicate protection plans and hence duplicated redundancy. In addition, crosslayer coordination is required for controlling and managing the multilayer protection (*e.g.*, to prevent duplicate actions). This, however, is still an open issue in research especially for the integrated scheme [52].

3.3.3 The case of overlay networks

Overlay networks are a possible solution to provide high end-to-end availability services on top of the IP layer. An overlay network is a logical set of nodes and links – typically, with its own set of protocols – configured on top of the IP layer of an existing infrastructure belonging to a single or multiple domains. Here, we consider the two types; Resilient Overlay Networks (RON) and Service Overlay Networks (SON). RON was first proposed by D. Andersen *et al.* [103]. Its basic idea is that overlay nodes actively probe each other to detect diverse paths in the underlaying layer. Each node maintains its own routing table based on the collected information. In the case of failure, the overlay network reroutes affected traffic through alternative paths found active. This is expected to take less time than rerouting at the native layer. The frequency and density of probing are required to be high in order to obtain the best routes, however, this increases network traffic. A trade-off between the two factors is addressed in [104]. The drawback of RON is that it does not take advantage of interlayer coordination in design nor in operation. Hence no guarantees on availability can be given. Unlike RON, SON typically reserves some capacity on the underlay network [105] and has a predefined logical topology layout [106]. The mapping of each overlay link (path) to underlay links is crucial in meeting both survivability and availability requirements. If availability levels on the underlay network are guaranteed to a SON operator, service availability on the upper layer can be computed accordingly. Hence, SON can be more advantageous to end users seeking specific levels of availability [107].

Generally, overlay networks can achieve high service availability by exploiting the redundancy in multiple layers measured by path diversity. An overlay network maintains its connectivity by rerouting the affected traffic at an underlay layer (if, dedicated or shared protection is given) or at the overlay layer. Thus it is capable of surviving a single failure and, potentially, multiple failures, but no hard guarantees on availability are given.

3.4 SUMMARY

Our research can be related to the existing literature in two problems, namely, the problem of supporting multiple levels of resilience and the problem of designing a survivable multilayer network. The main objective of this research is to combine these problems in order to support QoR classes in multilayer networks which has not been studied well as an integrated problem.

The problem of designing a survivable network supporting multiple levels of resilience is well studied in single layer networks. Essentially, the current approach is to take the physical network availability as a given and deploy redundancy and restoration techniques at various layers to provide QoR classes with different fault recovery capabilities and availabilities. Further supplementary research focuses mainly on improving resource efficiency and achieving high availability levels [108]. While these efforts have provided valuable solutions to the problem, most of their approaches suffer from the crosslayer mapping issues discussed in the literature as without full knowledge of the physical layer and the mappings between layers no hard guarantees on availability can be provided (*i.e.*, due to fault propagation). In addition, highest levels of availability are achievable only through adding more redundant paths (*e.g.*, enough spare capacity to reroute highest priority traffic after a second failure) which introduces scalability complexities. The closest approach to our proposal is Zhang et al. [49], which appeared shortly after our paper [109]. The authors try to optimize network availability by improving the availability of a subset of physical links via shielding. In addition, Botton et al. [110] study a network design problem with a subset of edges that for a given cost can be upgraded to be more reliable. They show that having a set of more reliable edges as a substitute to having edge-disjoint path-pairs can improve overall resource efficiency. These approaches, however, do not support resilience differentiation.

Also the crosslayer mapping part of the research is quite similar to the existing approaches [17, 18]. However, the essence of our constrained mapping model depends on creating multiple logical subnetworks with differentiated availability. Zhou et al. [19] had followed a similar approach by creating two or more logical networks such that one of them survives a failure. However, no availability classes are supported.

4.0 RESEARCH OVERVIEW

4.1 OVERVIEW

The core idea of this research is derived from the reliability theory of parallel components system: end-to-end availability of a parallel system is higher than the highest availability of the components. So by creating a high availability working path, we can ensure a minimum high level of the end-to-end availability. In [109], we design the spine to be a spanning tree so connections between any two nodes can be given working paths with high availability.

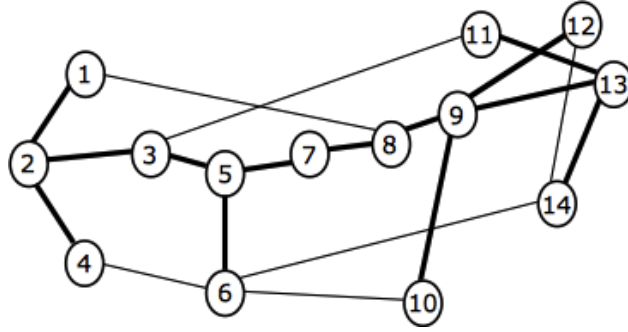


Figure 4.1: An example of a spine.

The basic idea of the spine is to design the network with a highly available subnetwork embedded in the physical network, as shown in Figure 4.1, where the bold links indicate the high availability links. This is achieved by managing to have the components on this subnetwork “the spine” with high availability, by reducing the *MTTR* and/or increasing the *MTBF* for these components. Afterwards, connections requiring high availability are routed

on-the-spine to achieve high availability working paths and protected, say, by a link disjoint dedicated backup path. A lower class can be routed on-the-spine with no protection or off-the-spine with any protection configuration (*e.g.*, DPP, SPP, $M:N$...etc). A best effort class is routed off-the-spine with no protection. Therefore, the spine provides a basis for differentiation between multiple QoR classes, and potentially can reduce the complexity of controlling and managing multiple QoR classes.

4.2 THESIS STATEMENT

This research aims to answer the following design questions:

Question 1. *Given an optical network with known links availabilities, options to improving links availabilities, the associated cost of the improvements, and a total budget, how to design the spine and select the best improvement option for each link in order to maximize the availability of the highest class of flows and widen the availability range of the network?*

Question 2. *Given a layered network with a spine embedded at the physical layer, and demands of different QoR classes, how to route logical demands and map logical links so that the network can distinguish QoR classes and provide differentiated availability?*

4.3 SCOPE AND ASSUMPTIONS

Throughout this research, we assume the following:

1. Links (or edges) in both physical and logical networks are undirected.
2. Availability computation considers only independent failures.
3. Flow availability is evaluated from the perspective of the physical network only. In addition, communication nodes are considered perfect (*i.e.*, $a_{node} = 1$) (see section 2.2.3).
4. Link availability is improvable (see section 2.2.4).
5. If protection is given, survivability is with respect to single failures.

6. Physical networks are real world networks. However, we assume logical networks are given. We generate synthetic random degree-constrained logical networks to represent the logical networks assuming that for each physical node there is an associated logical node.

Notwithstanding their importance, there are some related issues left outside the scope of this work:

1. The optimal design of the logical topology.
2. The accuracy of the availability model.
3. Reducing the computational complexity of the proposed algorithms.
4. The consideration of correlated failures in resilient network design (See [108]).

4.4 METHODOLOGY

To answer the aforementioned questions, we propose three design problems to study the problem of supporting differentiated classes of resilience in multilayer networks. The design problems are formulated as mathematical programming problems. The first design problem tries to answer the first question and aims to find the optimal spine. The proposed formulation is evaluated using real network topologies and the numerical results are evaluated in terms of connections end-to-end availability and the cost of the spine.

The second and third design problems attempt to answer the second question. Two models for routing logical demands and mapping logical links in multilayer networks are introduced. Both models consider differential crosslayer mapping in order to support differentiated classes of resilience. They differ in the layer where protection is given. Protection can be given at the physical layer as the in the first model or at the logical layer as in the second model.

4.5 LIST OF PUBLICATIONS

- [109] T. Gomes, D. Tipper, and A. Alashaikh, “A novel approach for ensuring high end-to-end availability: The spine concept,” in *Design of Reliable Communication Networks (DRCN), 2014 10th International Conference*, 2014, pp. 1–8.
- [111] A. Alashaikh, T. Gomes, and D. Tipper, “The spine concept for improving network availability,” *Computer Networks*, vol. 82, pp. 4–19, May 2015.
- [112] A. Alashaikh, D. Tipper, and T. Gomes, “Supporting differentiated resilience classes in multilayer networks,” in *2016 12th International Conference on the Design of Reliable Communication Networks (DRCN 2016)*, 2016, pp. 31–38.
- A. Alashaikh, D. Tipper, and T. Gomes, “Exploring the Logical Layer to Support Differentiated Resilience Classes in Multilayer Networks”, submitted to *Annals of Telecommunications*, [under review].

4.6 CONTRIBUTION

- In [109], the author of this dissertation participated in developing the approach, the model, and the analysis, and a minor contribution to the writing and the review of the paper.
- In [111], the author of this dissertation extended the previous work and included new analyses of related aspects, with advising and revision from the co-authors.
- In [112], the author of this dissertation developed the idea, experimental work, and writing with advising and revision from the co-authors.
- To the best of our knowledge, our proposal in [109, 111] was the first of kind that attempted to exploit link availability heterogeneity to provide high availability to connections. This is explained sections 5.1 to 5.3. In addition, we believe that this research

complements the previous work. Specifically, we relax the homogenous link availability assumption and utilize the spine concept in a multilayer scenario.

- We provide a different strategy to allocate budget dedicated to enhancing network resilience, which can result in potential saving and operational capability gains (*e.g.*, extremely high availability level, differentiation). This is presented in Section 5.5.
- We provide multiple models for differentiated routing at upper layer by utilizing two or more interleaved logical networks and availability-aware crosslayer mapping [112]. We consider two protection configurations, namely; protection at lower layer (Section 6.3) and protection at upper layer (Section 6.4).

5.0 THE SPINE CONCEPT

Given the frequency and the severity measures of failures in communications networks, discussed in Section 2.2.3, we believe that high availability must begin at the physical layer and work its way up the various layers. Here we explore an innovative technique termed the spine to improve overall availability and lay a basis for differentiation [113, 109]. In this chapter, we explain the core of our approach, *the spine concept*, and explore its feasibility. We also show methods to find suitable spines in physical layer networks and introduce our preliminary results.

5.1 DEFINITION

Our approach to provide high availability stems from the Brinbaum's importance measure. According to this measure, improving the link with the higher availability in parallel configuration yields the best overall availability [4]. To illustrate this theorem, consider the example in Figure 5.1. Assume we have a flow f routed over a WP and a BP and their availability is $A_f^{WP} = 0.99$ and $A_f^{BP} = 0.90$, respectively. Also assume we want to strengthen one or both of the links by adding some availability units Δa , with an option of $(A_f^{WP} + \Delta a)$, $(A_f^{BP} + \Delta a)$, or $(A_f^{WP} + \Delta a/2$ and $A_f^{BP} + \Delta a/2)$ to add these units. The end-to-end availability of flow f is based on its A_f^{WP} and A_f^{BP} , and it is calculated as a parallel configuration following equation (2.6). Figure 5.1 plots the overall end-to-end availability of flow f for the three options. It shows that improving A_f^{WP} only achieves better overall availability than any other option. From this, it is clear that having relatively highly available components in parallel combination can achieve better overall availability than homogenous availability

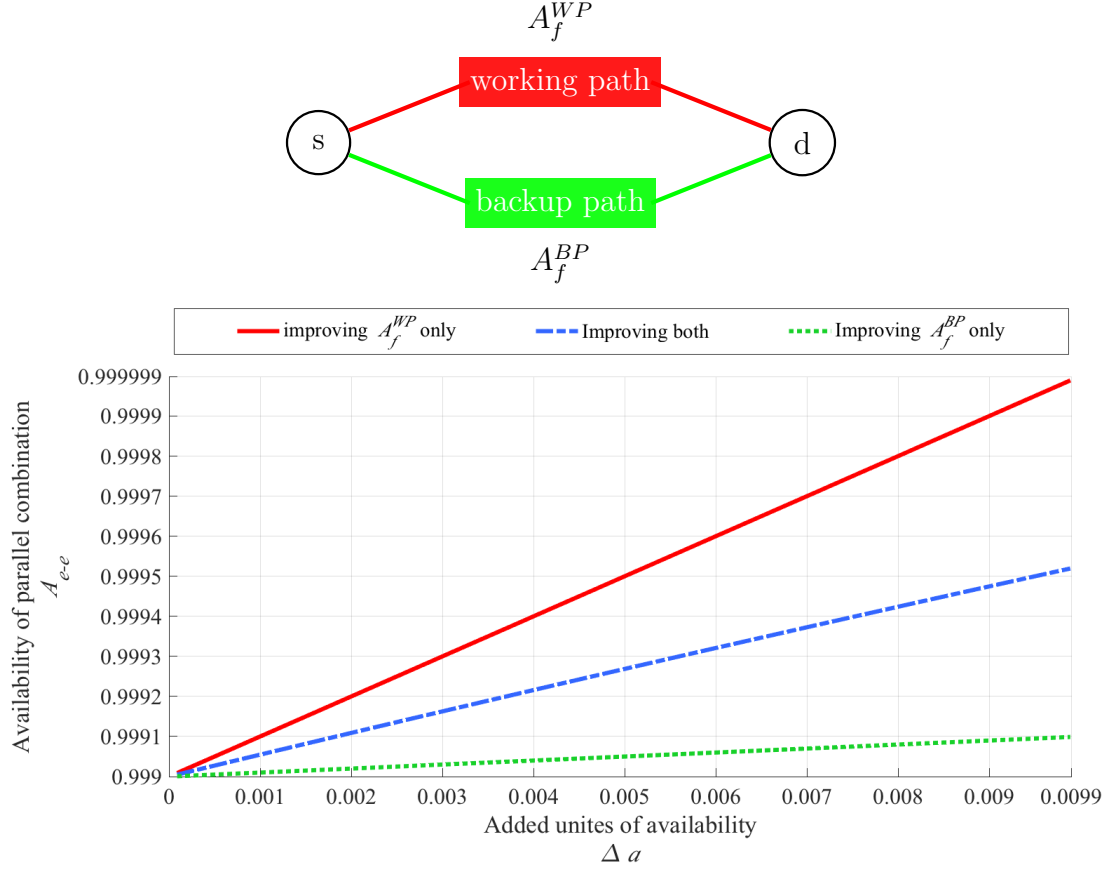


Figure 5.1: A plot shows that strengthen the strongest link in parallel configuration achieves the best overall end-to-end availability.

components. Adopting this concept, a network provider can allocate investment towards improving network reliability in an economic-efficient way.

Definition. *the spine is a substructure with comparatively higher availability embedded into the network at the physical layer to improve the overall network availability without substantial modifications to the topology.*

Our approach requires designing a network with heterogenous link availabilities such that a substructure of the network has relatively larger availability values. The high availability substructure portion of the network is termed the spine. The spine would connect those nodes with traffic needing a high level of availability and provide a basis for differentiated

classes of resilience. For example, the highest quality of resilience class traffic could be routed on the spine or use the spine as a backup path. The nodes, link interfaces and links on the network spine would have higher availability than the equipment that is not part of the spine. This provides *levels of availability differentiation at the physical level* which can be leveraged with restoration techniques, logical virtual network topology routing, cross layer mapping and other methods to further differentiate resilience classes and provide an extended range of availability guarantees. One can think of the spine approach as assuming a restoration method (path restoration) or a set of restoration methods (*i.e.*, no protection, shared backup path, dedicated backup path etc.) is to be used, then determining how should availabilities be assigned to the physical network components to best support the availability requirements.

5.1.1 The Spine Model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph of the physical network topology, where \mathcal{V} is the set of nodes and \mathcal{E} is the set of links. Also let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ be a subgraph of \mathcal{G} denoting the spine, where $\mathcal{V}_s \subseteq \mathcal{V}$ and $\mathcal{E}_s \subseteq \mathcal{E}$. The spine concept is to embed the spine \mathcal{G}_s with higher availability links and nodes on the physical network, such that $a_{v \in \mathcal{V}_s} \geq a_{v \notin \mathcal{V}_s}$ and $a_{e \in \mathcal{E}_s} \geq a_{e \notin \mathcal{E}_s}$, where a_v and a_e are the node and link availabilities, respectively. Here, however, we only consider link availabilities and assume the nodes are perfect (see section 4.3 and a remark in section 7.2). The spine layout design aims at improving overall availability and ensuring high end-to-end availability for all high availability communication services. In the general case, the spine could take any subgraph form as dictated by nodes and links availability and cost. Here, we assume that high availability communication service is needed between all $|\mathcal{V}| \times (|\mathcal{V}| - 1)/2$ node pairs (*i.e.*, full mesh). This means that the spine has to connect all nodes in \mathcal{G} , with a minimum number of links, and thus it takes the form of a spanning tree (*ST*) of graph \mathcal{G} .

5.1.2 Implementing Heterogenous Link Availabilities

The higher availability of the spine, in comparison to the non-spine part of the network, can be accomplished using a variety of techniques, as described in Section 2.2.4. For example,

on the spine more expensive equipment can be utilized that is arranged and configured to provide high availability (*e.g.*, hot standby line card, redundant fans, etc.) with redundant equipment deployed locally in parallel as needed (*e.g.*, hot standby fiber in physically diverse duct, etc.). Also, the equipment along the spine can be situated to increase the mean time to failure (MTTF) using a number of techniques such as longer back up power supplies, better heating/cooling, stronger outside cabinets, underground cabling instead of above ground, etc. In a similar vein, methods can be employed to reduce the mean time to repair (MTTR) along the spine. For instance, one can follow best practices and training procedures as determined by several government and trade organizations (*e.g.*, FCC, NRSC, NRIC, ATIS) and standards bodies (*e.g.*, ITU) [43, 46]. The operator can pre-position or relocate maintenance team centers, spare parts, equipments, software and test equipments along the spine. Similarly, the network operations center (NOC) can more closely monitor the spine portion of the network. Additionally, the operator can assign the most experienced staff to the operations, administration and management (OAM) of the spine portion of the network. Many of the methods above are employed in other critical infrastructures (*e.g.*, the power grid) and industries and studies show that the average MTTR can be reduced by 5-25% resulting in a significant improvement in the availability. Of course exactly which combination of techniques (hardware, equipment siting, workforce training, etc.) is adopted to improve the reliability of the spine will depend on the cost versus benefit structure of the network owner. Even using techniques to improve the MTTF and MTTR of links and nodes that comprise the spine, we assume additional protection, either end-to-end, segment or local [7] is needed to achieve the desired level of end-end availability for the most stringent QoR class.

5.2 EXPLORING THE SPINE

We explore the spine concept and its potential advantages through a series of simple examples.

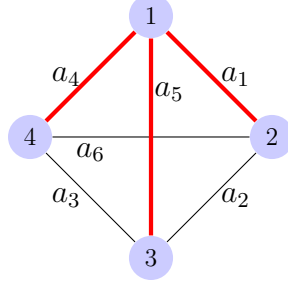


Figure 5.2: Full mesh network (thicker red lines denote spine).

5.2.1 Exploiting the Heterogeneity of Link Availability

Consider a full-mesh four node network as shown in Figure 5.2. To improve the end-to-end availability, we assume the network has the ability to employ disjoint working and backup path protection for each source-destination pair if desired. Thus each of the 12 source-destination pairs has a single hop direct working path (WP) and a disjoint two hop backup path (BP). Let A_S denote the average over all source-destination pairs of the end-to-end availability of a flow between a source-destination pair.

First we study the homogeneous case, $a_e = a$, for all $e \in \mathcal{E}$. The average system availability A_S is simply the parallel combination of the one hop working path and a two hop backup path which is given by:

$$A_S(a) = 1 - (1 - a)(1 - a^2) = -a^3 + a^2 + a \quad (5.1)$$

Now, let's consider the non-homogeneous edge availability case corresponding to the spine concept. We define a spanning tree as the spine consisting of edges 1, 5 and 4 as shown by the thicker red lines in Figure 5.2. Further we assume the availability of edges on the *spine* (a_1, a_4, a_5) are equal with value a_S and the availability of the edges off the spine (a_2, a_3, a_6) are equal with value a_O . Six of the node s - d pairs have a single hop WP on the spine and a two hop BP with one hop on the *spine*, so the corresponding availability is $1 - (1 - a_S)(1 - a_S a_O)$. The other six node pairs have a WP with two hops on the spine and a single hop BP off the spine, and the corresponding availability is $1 - (1 - a_S^2)(1 - a_O)$. So, the average end-to-end availability, as a function of a_S and a_O is: $A_S(a_S, a_O) = \frac{1}{12}(6(1 - (1 - a_S)(1 - a_S a_O)) + 6(1 -$

Table 5.1: Effect of varying ϵ on A_S and Downtime.

Case	A_S	Downtime (hours/year)
$a = .9, \epsilon = 0$.981	166.44
$a = .9, \epsilon = 0.09$.99712	25.23756
$a = .9 \epsilon = 0.099$.999701	2.61749

$(1 - a_S^2)(1 - a_O))$). If we assume that $a_S = a + \epsilon$ and $a_O = a - \epsilon$, then A_S can be shown to be

$$A_S(a, \epsilon) = -a^3 + (1 - \epsilon)a^2 + (1 + \epsilon)a + a\epsilon^2 + \epsilon^3 \quad (5.2)$$

Note, that since $a_S = a + \epsilon$ and $a_O = a - \epsilon$, then the average link availability and the sum of the link availabilities network wide are the same for the spine based network and the homogeneous case (*i.e.*, $\sum a_i = 6a$). We define δ as the difference in A_S between the spine and homogeneous scenarios, then $\delta = A_S(a, \epsilon) - A_S(a)$, which can be shown to be $\delta = \epsilon^3 + a\epsilon^2 + a\epsilon(1 - a)$, and $\delta > 0$ if $\epsilon > 0, a > 0$. Hence using edges with different availabilities results in larger average availability than using a homogeneous edge availability. So the spine has the potential to improve the average end-to-end availability.

In Table 5.1, we show numerical results of the effects of varying ϵ on A_S and the downtime per year for the four node full mesh network. From the table one can clearly see that embedding a spine with differential availability of the links has the potential to improve A_S . We also note that in the spine the different s - d node pairs do not always get the same level of availability. For example, when $\epsilon = 0.09$ the group of six s - d pairs with a single hop WP on the spine have end-to-end availability of 0.998, while the second group of s - d pairs with a two hop WP on the spine have end-to-end availability of 0.9962. Observe that both groups have an end-to-end availability greater than the uniform end-to-end availability provided by the $\epsilon = 0$ homogeneous case. An important point is that the choice of the spanning tree spine is not unique in maximizing A_S as selecting edges 1, 2, and 6 results in the same A_S . However, the choice of the spine is not arbitrary as selecting edges 1, 5, and 6 for the spine results in a lower A_S .

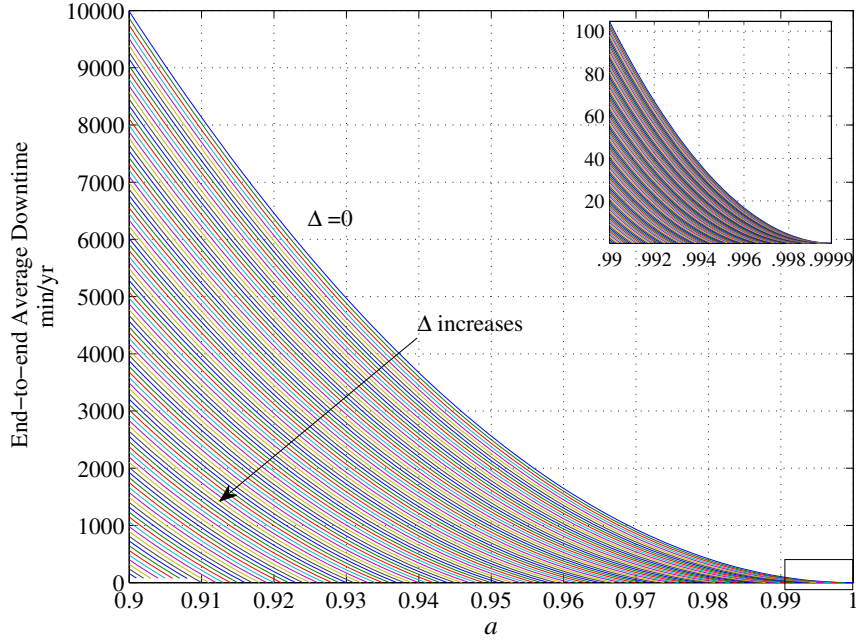


Figure 5.3: Average Downtime corresponding to A_S versus a and Δ .

Now we consider a slightly different scenario: given a topology, what is the effect of improving the availability of the components that make up the spine while leaving the rest of the network untouched? Specifically, we assume that $a_S = a + \Delta$ and $a_O = a$. Again considering the four node network in Figure 5.2 with the spine consisting of edges 1, 5 and 4 as shown by the thicker red lines, then A_S can be shown to be

$$A_S(a, \Delta) = \frac{1}{2}[-2a^3 + 2a^2 + 2a - (4a^2 - 3a - 1)\Delta - (2a - 1)\Delta^2]. \quad (5.3)$$

Figure 5.3 shows the average downtime in minutes per year for different a and Δ combinations. Each line corresponds to one Δ value, starting from zero (top line) and ending with 0.09999 in 100 steps, while varying the value of a . Thus each line shows how the average downtime decreases with increasing link availability a for a specific value of Δ . The inset figure in the top right corner is a magnification of the far right of the original figure. From Figure 5.3, a specific average downtime can be achieved either by improving a for all the six edges or by improving only the three edges on the spine to $a_S = a + \Delta$. For example,

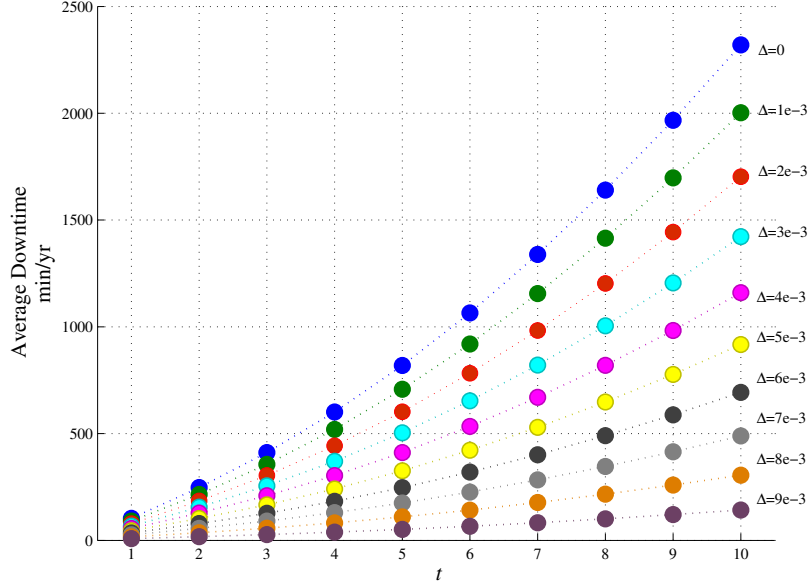
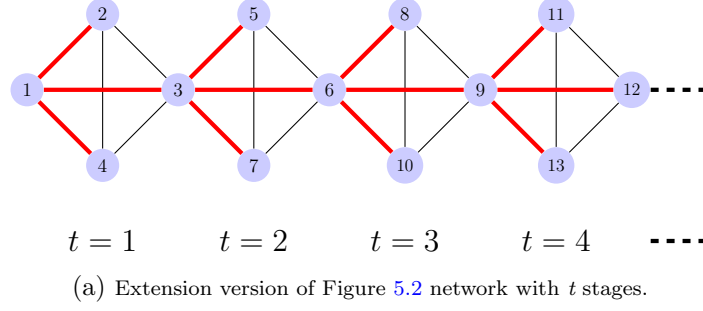
assume that initially all edges have $a = 0.92$ which results in an average flow downtime of about 6500 minutes/year. Say that we are targeting an average flow downtime of 2500 minutes/year. An option here is to improve a to $a = 0.95$ for all edges, or improve only the three edges on the spine to $a_S = a + \Delta \simeq 0.964$. Loosely speaking, in the first option the downtime of each edge is reduced by ≈ 263 min/yr and a total of $6 \times 263 = 1578$ min/yr, whereas in the second option a spine edge downtime is reduced by ≈ 385 min/yr and a total of $3 \times 385 = 1155$ min/yr. Hence, from a system downtime point of view it can be more effective to increase the availability of the spine components than to increase the availability of all the components in the network to achieve a target average flow downtime. This point will be further investigated in the subsequent sections.

5.2.2 Effect of Network Size

We expect the spine to be a more beneficial approach for large networks, where the longer paths between node pairs decreases the end-to-end flow availability significantly. Consider an extended version of the four node network of Figure 5.2, if we repeat the same structure with the same spine layout, we can produce a chain-like network as shown in Figure 5.4a with t stages (*i.e.*, t repetitions). The overall average availability A_S considering each WP on the spine and the BP as the corresponding min-hop edge-disjoint path can be derived as:

$$A_S = \frac{6}{n(n-1)} [t(a_S + a_S a_O - a_S^2 a_O + a_S^2 + a_O - a_S^2 a_O) + \sum_{r=2}^t (t-r+1) [a_S^r + a_S^r a_O^r - a_S^{2r} a_O^r + 2(a_S^{r+1} + a_S^{r-1} a_O^r - a_S^{2r} a_O^r)]] \quad (5.4)$$

where n is the number of nodes and t is the number of repetitions of the original network structure (*i.e.*, *stages*). As above we assume that $a_S = a + \Delta$ and $a_O = a$. In Figure 5.4b, we show the average downtime (in min/yr) for different t stage networks with a fixed $a = 0.99$ and Δ . A set of values are generated by varying Δ in steps of 0.001 over the range of 0 to 0.009. The top set of points in the figure shows the downtime for a homogenous case (with $\Delta = 0$). It can be seen that introducing differential link availability reduces the average downtime even for the larger networks as shown by the lower set of points. Note that for a specific Δ , the absolute change in average downtime is greater the larger the network.



(b) Average Downtime for $t = 1 : 10$, with $a = 0.99$, $\Delta = 0 : 9e - 3$ with $1e-3$ step size.

Figure 5.4: The effect of differential links availabilities on different sizes networks.

5.2.3 The Layout of The Spine

We note that the choice of a subgraph selected as the spine impacts the overall availability A_S . Consider the simple 5-node network shown in Figure 5.5 with two different spine layouts. The spine in the leftmost network is a star whereas the spine in the rightmost network is ring-like. As above, we assume that $a_S = a + \Delta$ and $a_O = a$ and one can show that for the leftmost star-like spine network

$$A_S(a, \Delta) = \frac{1}{5}[-a^4 - 4a^3 + 6a^2 + 4a - \Delta^4 + 4\Delta^3 + (2a^2 + 4a + 2)\Delta^2 - (4a^2 - 4a)\Delta]. \quad (5.5)$$

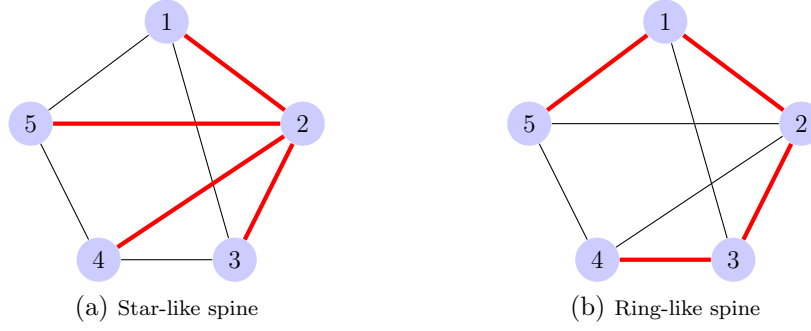


Figure 5.5: An example of a 5 nodes network with two spine designs.

Similarly for the rightmost ring-like spine network it can be shown that

$$A_S(a, \Delta) = \frac{1}{10}[a^4 - 5a^3 + 9a^2 + 8a - (3a - 1)\Delta^4 - (12a^2 - 4a - 2)\Delta^3 - (18a^3 - 6a^2 + a - 3)\Delta^2 - (12a^4 - 4a^3 + 8a^2 - 12a - 4)\Delta]. \quad (5.6)$$

Figure 5.6 (a) and (b) show plots of the downtime per year for different a and Δ for the star-like and ring-like spines respectively. Each line in the downtime plots corresponds to one Δ value, starting from zero (top line) and ending with 0.09999 in 100 steps, while varying the value of a . The spine in Figure 5.6a has much lower downtimes than the one in Figure 5.6b for a given a and Δ . Hence, the star-like spine is more efficient to reach a target downtime level and would be preferred to the ring-like spine.

In general, the choice of the spine is not unique and its selection not only impacts the average end-to-end availability A_S , but also the variability of the availability among $s - d$ pairs and the range of availabilities that can be selected by routing. For a realistic network topology, there will be many possible candidates to select the spine from and many factors come in to play in selecting the spine. For example, the length of the spine diameter (d_S) and the value of Δ are related. For any given flow, we may require the working path availability A_f^{WP} to be larger than the flow backup path availability A_f^{BP} . Thus $A_f^{WP} \geq A_f^{BP}$. Consequently, $a_S^{hc^{WP}} \geq a_O^{hc^{BP}}$, where hc^{WP} and hc^{BP} are the hop count for working and backup paths respectively. This relation should hold for all flows, and the worst case can be found for the flow with longest WP and shortest BP. The longest WP is obviously

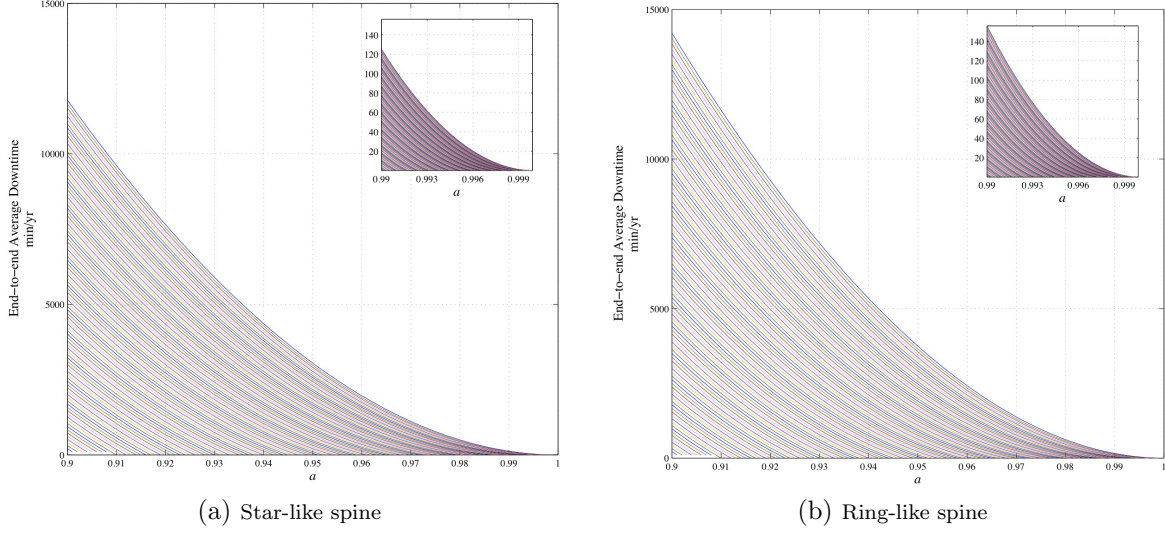


Figure 5.6: Average Downtime corresponding to A_S versus a and Δ for the two 5 nodes examples.

the diameter of the spine (dis), and the shortest can be one hop — this is a conservative approximation. Hence, if $a_S^{dis} \geq a_O$, then $dis \leq \frac{\ln a_O}{\ln a_S}$. This also constrains the minimum Δ value that can be applied to a given spine with a specific diameter, where $\Delta \geq (a^{1/dis} - a)$ must hold for the spine to meet the constraint. Hence, small values of Δ are worthwhile only for short dis . In the following section, we study how the properties of the network topology can be used to select a good spine.

5.3 EXPLORING *SPINE* SELECTION

Here, we consider how to select a good spine using minimum cost spanning trees, where the cost of using a link (or edge) was defined to take into account the edge betweenness centrality and the edge degree. The objective was to define the spine so that it would most likely include the edges that are important from the structural point of view of the network topology. However a spine is only considered admissible if an edge-disjoint WP and BP path can be calculated for each end-to-end $s - d$ node pair. Before presenting how we generate

and evaluate candidate spines, we detail our notation and provide some definitions.

5.3.1 Notation

Sets:

\mathcal{G}	network graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
\mathcal{V}	set of physical nodes in the graph.
\mathcal{E}	set of physical links in the graph (undirected edges).
\mathcal{E}_S	set of links in the spine.
\mathcal{G}_S	network subgraph defined by the spine, $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$.
\mathcal{F}	set of end-to-end flows

Indexes:

v	node index.
e	link (edge) index ($e \in \mathcal{E}$).
f	a bidirectional symmetric flow ($f \in \mathcal{F}$).
i, j	end nodes of a link ($i, j \in \mathcal{V}$).
s, d	end nodes of a flow ($s, d \in \mathcal{V}$).

Paths:

WP_f	Working Path for flow f .
BP_f	Backup Path for flow f .

Availability:

a_e	availability of link e .
A_f^{WP}	Working Path availability for flow f :

$$A_f^{WP} = \prod_{l \in WP} a_l \quad (5.7)$$

A_f^{BP}	Backup Path Availability for flow f (similar to equation (5.7)).
A_f	availability of flow f . Assuming WP_f and BP_f are edge-disjoint, $A_f = 1 - (1 - A_f^{WP})(1 - A_f^{BP})$.

A_S^{WP}	average value of A_f^{WP} when WP on the spine.
A_S^{BP}	average value of A_f^{BP} when WP on the spine.
A_S	average value of A_f when the WP on the spine.

Performance and Structural measures:

eb_e The edge e betweenness centrality which is determined from:

$$eb_e = \frac{2}{|\mathcal{V}|(|\mathcal{V}| - 1)} \sum_{s,d \in \mathcal{V}} \frac{\sigma(s,d|e)}{\sigma(s,d)} \quad (5.8)$$

where $\sigma(s,d)$ is the number of shortest paths between nodes s and d and $\sigma(s,d|e)$ is the number of those paths that use edge e .

eb_S (eb_G) The average value of eb_e in \mathcal{G}_S (\mathcal{G}), that is considering only the edges in \mathcal{E}_S (\mathcal{E}).

h_S (h_G) The average shortest paths in $\mathcal{G}_S(\mathcal{G})$ (based on the hop-count only, link distance is not considered).

ed_e is the degree of edge e , defined as the sum of the degree of the edge's end nodes.

ed_S (ed_G) The average of ed_l over all edges in $\mathcal{G}_S(\mathcal{G})$.

di_S (di_G) The spine diameter, that is the length (hops) of the longest shortest path in $\mathcal{G}_S(\mathcal{G})$.

c_e cost of using edge e .

H_S The total number of hops (*i.e.*, links) used by WPs and BPs of all flows in \mathcal{F} when the WPs have to be in the spine.

H_G The total number of hops used by WPs and BPs of all flows in \mathcal{F} when the WPs do not have to be in the spine.

$\Delta H\%$

$$\Delta H\% = \frac{H_S - H_G}{H_G} \times 100 \quad (5.9)$$

5.3.2 Generating Candidate Spines

To generate candidate spines, we used Kruskal's minimum spanning tree (MST) algorithm with the cost of the edges c_l^i defined as a weighted combination related to the edge betweenness centrality and the edge degree. The costs of the edges c_l^i , $i \in \{A, B, C, D\}$ were determined as follows:

- Case A: for a given $\alpha > 0$, the larger the edge degree and the larger the edge betweenness centrality, the smaller the cost of edge l :

$$c_e^A = (1 - \alpha) \frac{(\min_e ed_e)}{ed_e} + \alpha \frac{(\min_e eb_e)}{eb_e} \quad (5.10)$$

- Case B: for a given $\alpha > 0$, the larger the edge degree and the smaller the edge betweenness centrality, the smaller the cost of edge l :

$$c_e^B = (1 - \alpha) \frac{(\min_e ed_e)}{ed_e} + \alpha \frac{eb_e}{(\max_e eb_e)} \quad (5.11)$$

- Case C: for a given $\alpha > 0$, the smaller the edge degree and the larger the edge betweenness centrality, the smaller the cost of edge l :

$$c_e^C = (1 - \alpha) \frac{ed_e}{(\max_e ed_e)} + \alpha \frac{(\min_e eb_e)}{eb_e} \quad (5.12)$$

- Case D: for a given $\alpha > 0$, the smaller edge degree and the smaller edge betweenness centrality, the smaller the cost of edge l :

$$c_e^D = (1 - \alpha) \frac{ed_e}{(\max_e ed_e)} + \alpha \frac{eb_e}{(\max_e eb_e)} \quad (5.13)$$

In all cases (*i.e.*, A-D) the weight α was varied from zero to one in increments of 0.1, and Kruskal's algorithm [114] was used for generating a MST for each value of α . If the resulting MST was equal to one previously obtained, it was dropped. Also, if the obtained MST (spine) did not allow for all WPs in the spine to be protected by an edge disjoint BP, the MST was dropped. In this case, the set (X) of all the common edges between a WP and its BP for all s - d pairs was collected. Then, sequentially, each combination (1 to $|X|$) of the common edges was temporarily removed from the graph and Kruskal's algorithm was again used, until either an admissible MST was obtained or the network became disconnected.

Table 5.2: Test Network Topology Data.

Network	$ \mathcal{V} $	$ \mathcal{E} $	$\frac{ \mathcal{E} }{ \mathcal{V} }$	$eb_{\mathcal{G}}$	$ed_{\mathcal{G}}$	$h_{\mathcal{G}}$	$di_{\mathcal{G}}$	$H_{\mathcal{G}}$
Polska	12	18	1.50	0.1187	6.3333	2.1364	4	356
NSF	14	19	1.36	0.1180	5.7895	2.2418	4	559
Spain	14	22	1.57	0.1034	6.8182	2.2747	5	526
Italia14	14	29	2.07	0.0644	10.3448	1.8681	3	425
EPAN	16	23	1.44	0.1149	6.0870	2.6417	6	806
Italia	32	69	2.16	0.0425	9.4493	2.9315	6	3378

We assume all links on the spine have the same availability $a_e = a_S \forall l \in \mathcal{S}$ and all links off the spine have the same availability $a_e = a_O \forall l \in \mathcal{E} - \mathcal{E}_S$. The WPs were routed entirely on the spine while each BP, edge-disjoint with the corresponding WP, was calculated with high edge cost on the spine (*i.e.*, to avoid routing the BP on the spine). Specifically, prior to determining each BP, the cost of the edges of the protected WP was defined equal to a sufficiently large number, the cost of the rest of the edges in the spine was increased and the remaining edges had their cost changed to one. This way the BP is maximally edge-disjoint with the corresponding WP, while avoiding the edges in the spine (if possible). The common edges were used to generate new candidate MSTs as described above. The set of candidate spines were evaluated considering the metrics: A_S , A_S^{WP} , h_S , di_S , $\min_f A_f$, H_S and $a'_e \forall e \in \mathcal{E}$ the uniform edge availability required to achieve the same A_S as the spine based solution.

5.3.3 Numerical Results

Here we present sample results for a set of network topologies often adopted in the literature, Polska, NSF [115], Spain [116], Italia14 [117], EPAN16 [118], and Italia [119]. In Table 5.2, data on the topologies of the test networks is given. In the results presented here, we use $a_O = 0.99$ and $a_S = 0.999$ and a step size of 0.0001 in determining a'_e . Boldface is used in the table of numerical results to make the corresponding maximum (minimum) values in some

Table 5.3: Numerical Results for Heuristics.

Network	eq. for c_e^i	α	A_S^{WP}	A_S	$\min_f A_f$	eb_S	ed_S	h_S	di_S	H_S	a'_e
Polska	(5.11)	0.1	0.99734	0.9999294	0.9997033	0.2424	5.2727	2.6667	5	391	0.9969
	(5.12)	0.6	0.99660	0.9999480	0.9998543	0.3099	4.3636	3.4091	8	425	0.9973
	(5.12)	0.8	0.99714	0.9999398	0.9997890	0.2603	4.9091	2.8636	6	395	0.9971
NSF	(5.10)	0.8	0.99685	0.9999382	0.9997986	0.2426	4.7692	3.1538	6	593	0.9973
	(5.12)	0.8	0.99643	0.9999405	0.9997527	0.2747	4.6154	3.5714	7	634	0.9974
	(5.11)	0.1	0.99662	0.9999347	0.9998518	0.2604	4.6154	3.3846	7	616	0.9973
	(5.10)	0.8	0.99672	0.9999391	0.9997859	0.2527	4.6154	3.2857	7	603	0.9973
Spain	(5.11)	0.3	0.99717	0.9999063	0.999651	0.2181	5.6923	2.8352	6	608	0.9968
	(5.12)	0.4	0.99611	0.9999226	0.999725	0.3001	4.3077	3.9011	9	626	0.9972
	(5.11)	0.4	0.99642	0.9999066	0.999621	0.2756	4.6154	3.5824	9	649	0.9968
Italia14	(5.10) or (5.11)	0.0	0.99779	0.9999374	0.9997964	0.1699	9.0769	2.2088	3	456	0.9965
	(5.10) or (5.12)	0.9	0.99764	0.9999446	0.9998503	0.1817	7.6923	2.3626	4	462	0.9972
	(5.12)	0.8	0.99755	0.9999416	0.9998426	0.1885	7.0769	2.4505	4	473	0.9972
EPAN16	(5.10)	0.1	0.99670	0.9999149	0.9997116	0.2200	5.0667	3.3000	7	838	0.9973
	(5.12)	0.5	0.99642	0.9999218	0.9997412	0.2389	4.6667	3.5833	8	890	0.9974
	(5.11)	0.3	0.99642	0.9999157	0.9997116	0.2389	4.8000	3.5833	8	861	0.9973
Italia	(5.10)	0.1	0.99623	0.9998015	0.9992670	0.1217	6.9677	3.7742	8	4367	0.9960
	(5.12) or (5.13)	0	0.99348	0.9998115	0.9993595	0.2110	4.5806	6.5423	15	4837	0.9961
	(5.10)	0.5	0.99616	0.9998041	0.9992670	0.1241	6.9677	3.8468	8	4341	0.9960

columns more visible, depending on which was considered relevant for that column.

Table 5.3 shows numerical results for the six networks studied where the spines were found using the cost functions above. First we consider the results for the Polska network. In Table 5.3 the first row corresponds to the MSTs with largest A_S^{WP} , and the second row corresponds to the MST with the largest A_S , using equations (5.11)-(5.12). The corresponding spines are shown in Figures 5.7a and 5.7b, respectively. It can be seen that the largest A_S corresponds to a MST that presents a large diameter (twice the network diameter di_G), while the MST with the largest A_S^{WP} has a diameter of only 5. Also note that the row with maximum A_S^{WP} has the smallest H_S . In the third row we present a spine with metrics in between the two above, the spine has a diameter of 6 and is shown in Figure 5.7c.

Considering the spines with the largest A_S (second row in Table 5.3), it is worth noting that the corresponding value of a'_e (0.9973) is larger than the average value of the edges availability (a_e) considering the spine 0.9955. This confirms the results in Section 5.2. In the

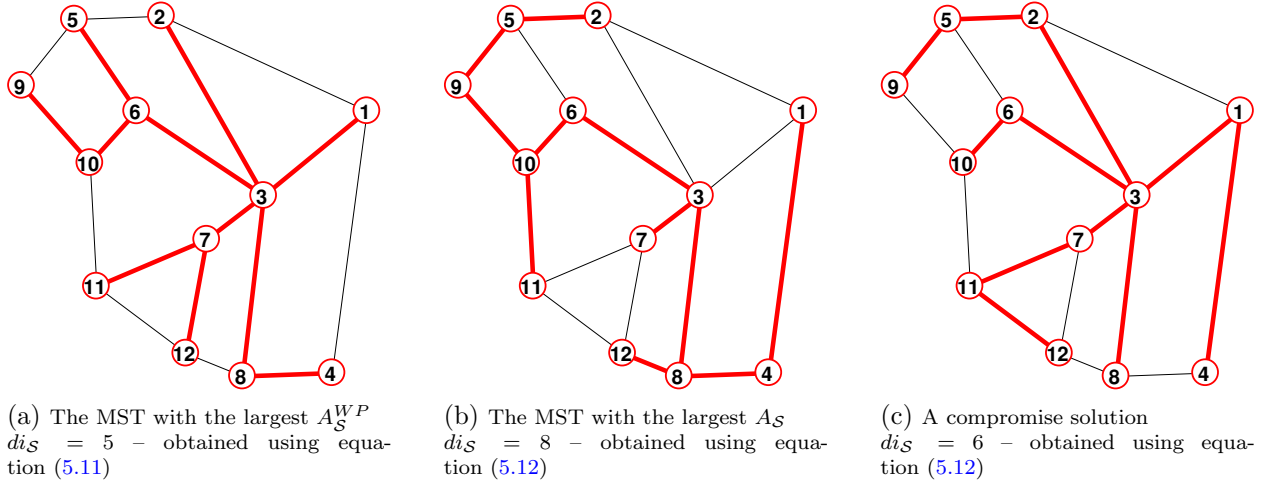


Figure 5.7: The MSTs obtained by heuristics for Polska network – red/thicker lines represent the spine.

second row of Table 5.3 one can also find the largest obtained value for $\min_f A_f$ (considering 1+1 protection).

The NSF network results in Table 5.3 list in the fourth row the values corresponding to the spine with largest A_S^{WP} , in the fifth row the values corresponding to the spine with largest A_S , in the sixth row the values corresponding to the spine with the largest value for $\min_f A_f$ and finally in the seventh row, a compromise in between solution. As in the case of the Polska network a larger A_S can be obtained at the cost of a larger spine diameter. In this case the spine that results in the maximal value for $\min_f A_f$ does not coincide with the spine with the largest A_S , and it also has a lower A_S than the corresponding value in the fourth row (row of maximum A_S^{WP}). The compromise solution has A_S^{WP} , A_S and $\min_f A_f$ in the interval defined by the corresponding values in rows four and five of Table 5.3.

The Spain network results in Table 5.3 list in the eighth row the values corresponding to the spine with largest A_S^{WP} with a diameter of 6, and in the ninth row the values corresponding to the spine with largest A_S , but it has a longer diameter. The two spines are followed by a compromise solution in the tenth row, the spine with second largest A_S^{WP} . The spine that results in the maximal value for $\min_f A_f$ happens to have a comparatively lower

availability values than the three spines listed here.

For the Italia14 network, the spine with largest A_S^{WP} has a diameter of 3, the same as in the full graph. The spine with largest A_S coincides with the spine with maximum $\min_f A_f$. The compromise solution spine has the second largest A_S .

The results obtained for the EPAN16 network (shown in line fourteen to sixteen in Table 5.3) are similar to those obtained for the Polska and NSF networks. The proposed compromise solution has A_S equal to 0.9999157, slightly larger than the corresponding value in row thirteen of Table 5.3, while presenting the same $\min_f A_f$.

In the case of the Italia network, the largest value for A_S^{WP} was obtained considering the costs given by equation (5.10) with α equal to 0.1, and is presented in row seventeen of Table 5.3. A compromise solution can be found in the last line of the table. It achieves the $\min_f A_f$ value shown in line eleven and has a larger A_S than the corresponding value found in that same row. The spine resulting in the largest A_S (and $\min_f A_f$) was obtained twice (see row eighteen), because when α is zero, the cost given by equations (5.12) and (5.13) is equal to $ed_e / \max_e ed_e$. It can also be observed that the required uniform edge availability (a'_e), to achieve the value of A_S in Table 5.3, is 0.9961, while using a spine this can be achieved with an average of 0.99404. Note that in the case of Italia network the minimal value obtained for H_S does not correspond to the spine with maximal A_S^{WP} (as was the case of the previous networks).

Overall, from the numerical results, it was observed that for each type of cost (5.10)-(5.13) the MST with the largest average WP availability A_S^{WP} often also corresponds to the MST with the smallest average shortest path h_S , the smallest H_S , the smallest diameter di_S and smallest average edge betweenness centrality eb_S , and with the largest average edge degree ed_S . However the MST that corresponds to the largest A_S^{WP} rarely coincides with the MST with the largest value for A_S . Nevertheless the MST that maximizes A_S (for each type of cost) tends to present, a small h_S , di_S , eb_S (although these are usually larger than the corresponding values for the MST that maximizes A_S^{WP}), and a large ed_S (although usually smaller than the MST that maximizes A_S^{WP}). Also, the results from the tested networks seem to indicate that maximizing A_S^{WP} does not maximize $\min_f A_f$.

5.4 CONSIDERING ALL SPANNING TREES IN DETERMINING THE SPINE

In this section, we study the metrics used to evaluate spine solutions over the space of all spanning trees in order to see their behavior and gain insight into spine selection.

5.4.1 Generating All Spanning Trees

The number of spanning trees (ST) in a connected graph \mathcal{G} can be quite large even for small $|\mathcal{V}|$ and $|\mathcal{E}|$. The exact number of STs in a graph can be related to the Laplacian spectrum of the graph [120] as follows. Let A denote the $|\mathcal{V}| \times |\mathcal{V}|$ adjacency matrix of a graph, where $a_{ij} = 1$ if and only if there is a link between node i and node j , otherwise $a_{ij} = 0$. The degree matrix D is a $|\mathcal{V}| \times |\mathcal{V}|$ matrix with the node degree placed along the diagonal (*i.e.*, d_{ii} = number of adjacent nodes of i) and zero every where else. The Laplacian matrix L of a graph is defined as $A - D$ and the eigenvalues $\lambda_i, i = 1, 2, \dots, |\mathcal{V}|$ of L form the Laplacian spectrum. It has been shown in the algebraic graph theory literature [120] that the number of spanning trees in a graph can be determined from the Laplacian spectrum by:

$$\text{No. of Spanning Trees in } \mathcal{G} = \frac{1}{n} \prod_{i, i > 1} \lambda_i \quad (5.14)$$

Table 5.4 shows the number of spanning trees for the networks studied here. One can clearly see, that for even modest size networks such as EPAN16, the number of spanning trees is quite large.

In order to generate all STs, we use Prim's algorithm to determine a ST implementing a binary code of size $(2^{|\mathcal{E}|})$ with $|\mathcal{E}|$ digits each corresponding to a specific link on the graph, with value 1 if the link is on the spine and 0 if not. Then, we run a counter from $(2^{|\mathcal{V}|} - 1)$ to $(2^{|\mathcal{E}|} - 1)$ to enumerate all possible combinations of links on a spine. Each generated combination of links that constructs a valid ST is saved to be further tested. A valid ST is verified by checking that the sum of all columns on the adjacency matrix of the spine is greater or equal to one and the number of links is $|\mathcal{V}| - 1$. Once all spanning trees are created, for each ST we route the WPs for all $s - d$ pairs on the spine while BPs are routed such

Table 5.4: Number of spanning trees for sample networks.

\mathcal{G}	Network	$ \mathcal{V} $	$ \mathcal{E} $	$\frac{ \mathcal{E} }{ \mathcal{V} }$	No. of ST	No. of <i>valid</i> spines
1	Polska	12	18	1.50	5161	1862
2	NSF	14	19	1.36	5862	1466
3	Spain	14	22	1.57	40436	22037
4	Italia14	14	29	2.07	1194812	not attempted
3	EPAN16	16	23	1.44	43720	7535
4	Italia	32	69	2.16	53.3E+14	not attempted

they avoid the spine if possible but constrained to be fully disjoint from the corresponding flow's WP. The ST that allows an edge-disjoint BP for each WP is considered a valid spine. Table 5.4 shows, in the last column, the number of valid spines for the networks under study.

5.4.2 Numerical Results

We studied the Polska, NSF, Spain, and EPAN16 networks by generating all STs and routing all s - d flows with disjoint protection. As in the previous sections all links on the spine have the same availability a_S and all links off the spine have the same availability a_O . Here we use $a_S = 0.999$ and $a_O = 0.99$. We evaluate the results using the same performance and structural measures as in the previous section. Table 5.5 shows the results for the networks considered. For each network in the table, the first row corresponds to the ST with the largest A_S^{WP} , the second row the ST corresponding to the largest A_S and the subsequent rows can be either the ST with largest $\min_f A_f$ or a compromise solution between the results of the first two rows. In addition to the evaluation metrics considered in Table 5.3, we include $\Delta H\%$, the percentage of increase in path lengths required by the spine, H_S , compared to the full graph, H_G , as given in Equation (5.9).

We find the observations from Section 5.3.3 regarding the coincidence between A_S and relatively to average, low eb_S , h_S , dis_S , and high ed_S to hold. Similarly, the correspondence of

Table 5.5: Results considering all spanning trees.

Network	A_S^{WP}	A_S	$\min_f A_f$	eb_S	ed_S	h_S	dis	H_S	$\Delta H\%$
Polska	0.99734	0.9999322	0.9997554	0.2424	5.2727	2.6667	5	385	8.2
	0.99660	0.9999480	0.9998543	0.3099	4.3636	3.4091	8	425	19.4
	0.99732	0.9999440	0.9997967	0.2438	5.4545	2.6818	5	379	6.0
	0.99654	0.9999442	0.9998809	0.3154	4.1818	3.4697	8	411	15.5
NSF	0.99687	0.9999390	0.9997642	0.2409	4.6154	3.1319	6	583	4.3
	0.99665	0.9999424	0.9998518	0.2578	4.6154	3.3516	7	604	8.1
	0.99637	0.9999415	0.9998736	0.2798	4.4615	3.6374	9	639	13.0
Spain	0.99721	0.9999233	0.9997212	0.2147	5.6923	2.7912	5	560	6.5
	0.99709	0.9999429	0.9997938	0.2240	5.5385	2.9121	6	586	11.4
	0.99652	0.9999320	0.9998469	0.2680	4.6154	3.4835	7	598	13.7
EPAN16	0.99670	0.9999149	0.9997116	0.2200	5.0667	3.3000	7	838	4.0
	0.99649	0.9999254	0.9997724	0.2344	4.6667	3.5167	8	869	7.8
	0.99552	0.9999163	0.9998128	0.2994	4.1333	4.4917	11	936	16.1

A_S^{WP} on the spine with minimum dis and h_S (shown in boldface in Table 5.5) and relatively large ed_S . Figures 5.8-5.11 show, for each network, A_S^{WP} , A_S , and $\min A_f$ for all STs examined with the results sorted from largest to smallest. On each plot the right side scale is the corresponding unavailability. Also, we mapped (shown as dots) the STs obtained from the heuristics on the plots. Note that the figures have fairly consistent behavior in terms of the shapes of the curves across the four networks. For A_S^{WP} there appears to be a relatively small set (in comparison to the number of STs) of STs with the largest values, whereas for A_S there is a larger percentage of STs with reasonably high values. But these values corresponds to different $\min A_f$. In Polska network, the difference between the largest and smallest $\min A_f$ values is 2.89e-4, compared to 6.35e-5 for A_S , which indicates that A_S values range is narrower. It might be more appropriate to consider the *minimum* flow availability(A_f) instead of A_S , to ensure that each flow can be given a route with highest possible end-to-end availability.

Comparing with the heuristics of Section 5.3.3, the largest value obtained from the heuristics for A_S^{WP} in the Polska network, shown in Table 5.3, coincides with the maximum found

over all STs in Table 5.5 – there are multiple STs of the same A_S^{WP} value. For the EPAN16 network, the heuristics in Section 5.3.3 calculated a spine with A_S^{WP} equals to 0.99670 which also coincides with the maximum found over all STs in Table 5.5. Regarding the case of the NSF network, the maximum value obtained by the heuristics for A_S^{WP} (0.99685) was also close to the maximum 0.99687 in Table 5.5. Similarly, the spine found by heuristics for the Spain network for A_S^{WP} was 0.99717 compared to 0.99721 across all STs.

In terms of A_S , for the Polska network the maximum value found from the heuristics (0.9999480) is actually the maximum over all STs in Table 5.5. Similarly, the heuristic in Section 5.3.3 managed to generate a spine for the NSF network, such that the resulting A_S

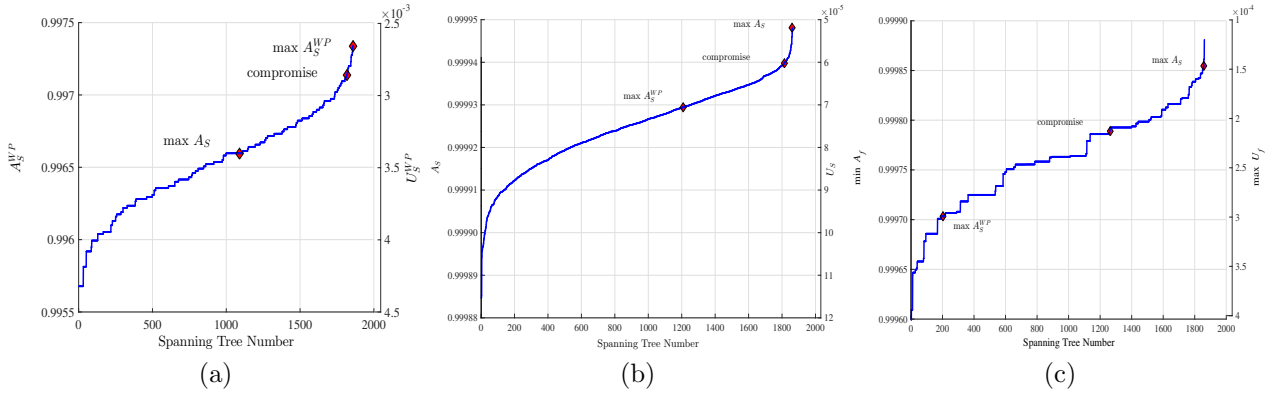


Figure 5.8: A_S^{WP} , A_S & $\min-A_f$ calculated over all spanning trees for Polska Network.

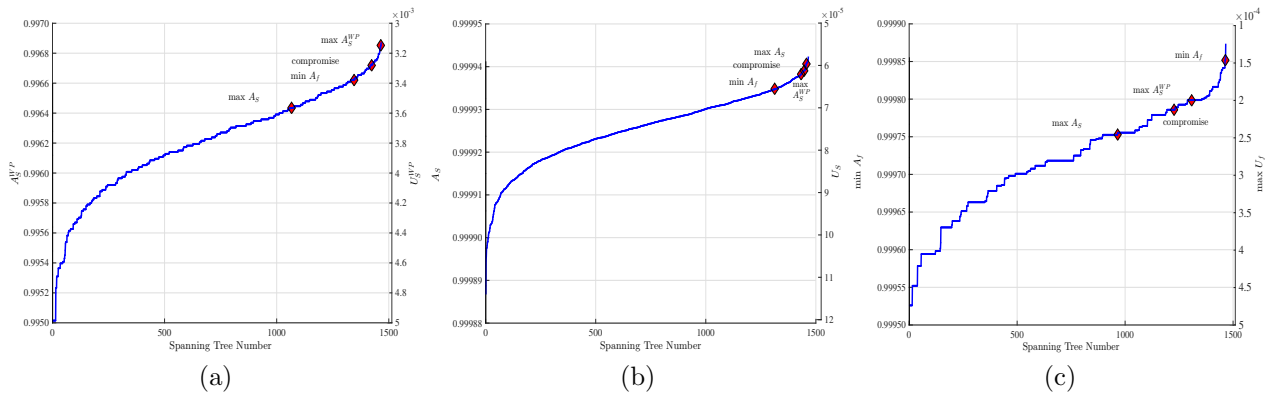


Figure 5.9: A_S^{WP} , A_S & $\min-A_f$ calculated over all spanning trees for NSF Network.

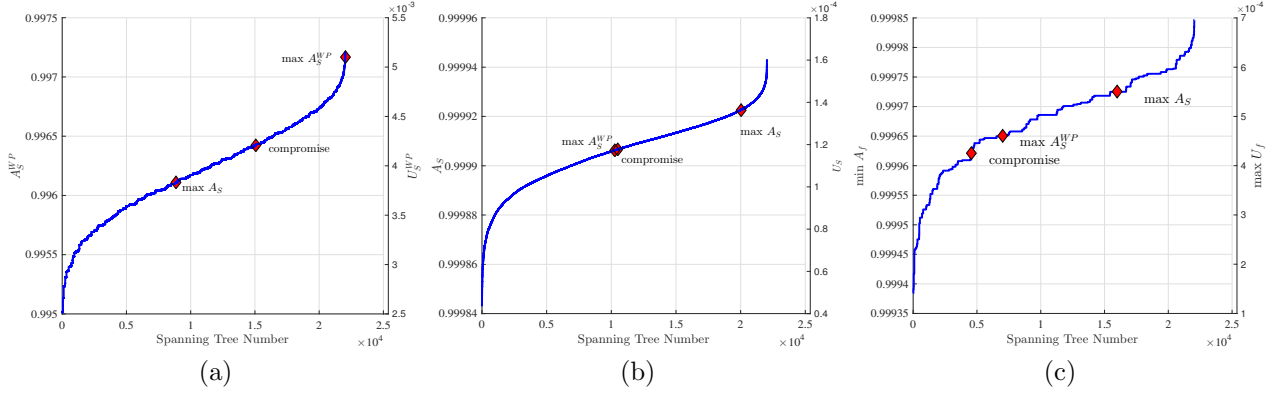


Figure 5.10: A_S^{WP} , A_S & $\min A_f$ calculated over all spanning trees for Spain Network.

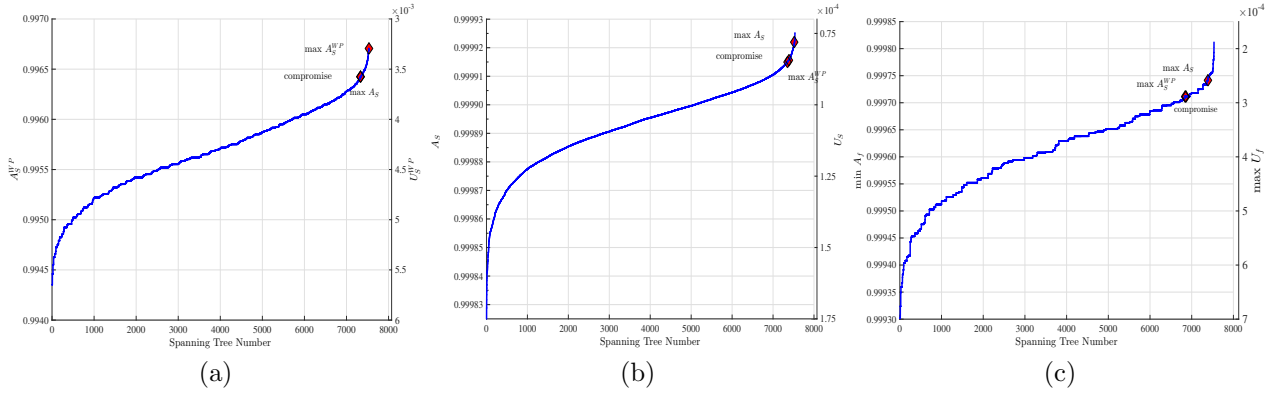


Figure 5.11: A_S^{WP} , A_S & $\min A_f$ calculated over all spanning trees for EPAN16 Network.

has the first 5 digits correct (according to Tables 5.3 and 5.5). In the case of the EPAN16 network, the first 5 digits match too. In the case of the Spain network, there is a $2e-5$ difference in A_S between the spine obtained from heuristics and all STs. Overall these numerical results seem to indicate that the heuristics from Section 5.3.3 work reasonably well in identifying a viable spine for a network topology.

Regarding path lengths, Table 5.5 (last two columns) shows the total number of links/hops utilized by the network, H_S , when the spine is considered. This is compared to the total number hops used when the link-disjoint path pair is calculated using shortest path pair (*i.e.*, *min-min*). It is clear from the results that the spine incurs more resources. Despite

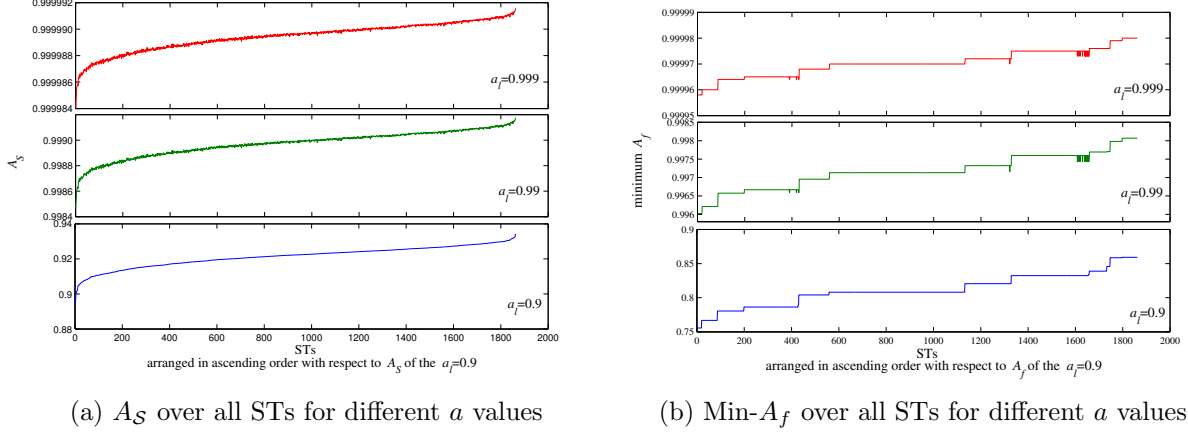
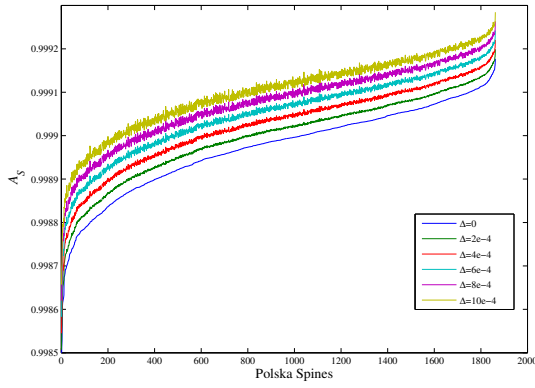


Figure 5.12: The effect of varying a in Polska network.

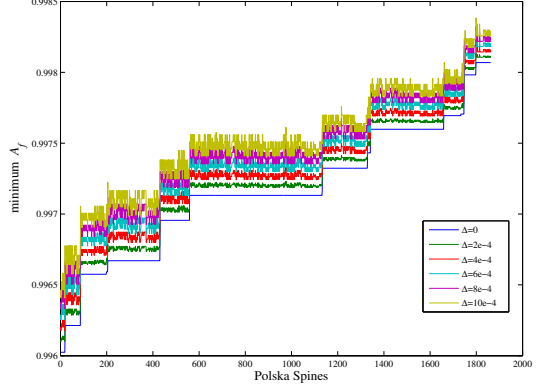
this, the increase in path lengths $\Delta H\%$ can be as low as 6%, 4.3%, 6.5%, and 4.0% for Polska, NSF, Spain, and EPAN16 networks, respectively. These low percentages correspond to spines with short diameters whereas spines with longer diameters result in larger ΔH_S 's. For the spines with longer diameters, the increase in H_S is mainly due to a large increase in WPs length accompanied with a slight or no decrease in BPs length.

5.4.3 Sensitivity Analysis

Homogenous Scenario In our previous numerical analysis, we considered $a = 0.99$ and $\Delta = 0.009$ (*i.e.*, $a_O = 0.99$, $a_S = 0.999$). Here we examine how the results would change if we considered different values of a and Δ . Figure 5.12 (a) and (b) shows the availability values (A_S and minimum A_f) for all STs in the Polska network using 3 different values of $a = \{0.9, 0.99, 0.999\}$ with $\Delta = 0$. By visual inspection, we can see that the general pattern for the ordered STs remain unchanged regardless of the metric A_S or minimum A_f . Only slight changes in terms of the ranking of the individual STs from best to worst occur in both measures. For varying Δ , Figure 5.13 shows the results for A_S and minimum A_f using 5 different Δ values with $a = 0.99$. When Δ changes, we see a consistent pattern for the values. However, more variation is noticed. This means that the ranking of the STs (from

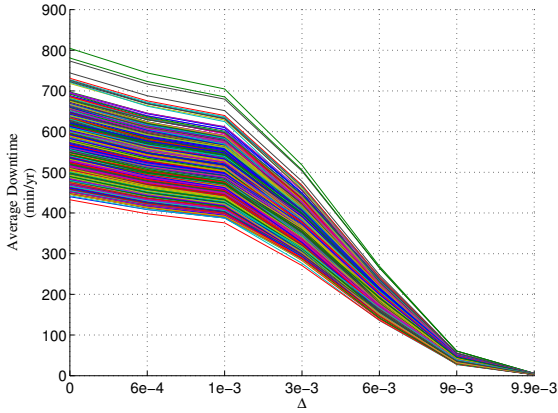


(a) A_S over all STs for different Δ values

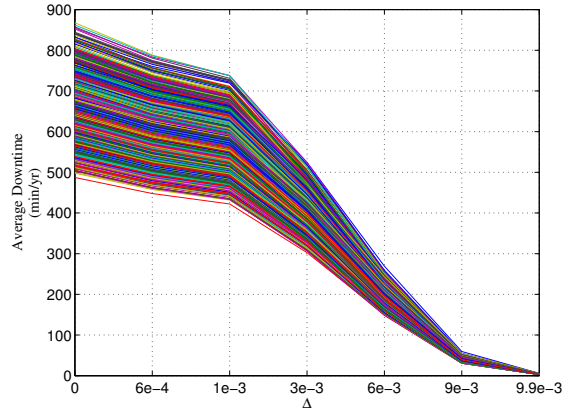


(b) Min- A_f over all STs for different Δ values

Figure 5.13: The effect of varying Δ in Polska network.



(a) Polska Network



(b) NSF Network

Figure 5.14: Average Downtime corresponding to A_S for all STs at different Δ values.

best to worst) with respect to a specific measure would slightly change as Δ changes and an ST might exchange its rank with another ST within its close range. In addition, for each value of Δ there can be different best/worst STs. Figure 5.14 visualizes this behavior, especially at large Δ values. For example, in Figure 5.14a at $\Delta = 6e - 3$ the upper three STs (corresponds to worst A_S) exchange their positions after the merging point as the variation

of the values diminishes. This is also true for the NSF network as shown in Figure 5.14b.

Heterogenous Scenario So far, we considered different values for a and Δ but we also assume that all links on the spine have the same availability $a_e = a + \Delta \forall e \in \mathcal{E}_S$ and all links off the spine have the same availability $a_e = a \forall l \in \mathcal{E} - \mathcal{E}_S$. However, typically, links on a network would have different availabilities. In this part, we relax our assumption of considering homogenous link availability. We consider a distance-based link availability model found in [7]. The link availability is calculated as $a_e = a_c \times a_t$ where a_t is the product of cable-ends equipments (*i.e.*, OXC, ADM etc...), and a_c is the fiber cable availability that can be calculated from:

$$a_c = 1 - \frac{MTTR}{MTBF} \quad (5.15)$$

$$MTBF_{hrs} = \frac{CC \times 365 \times 24}{\text{cable length}_{km}} \quad (5.16)$$

where $MTBF$ and $MTTR$ are the mean time between failures and mean time to repair in hours, respectively. CC is the cable cut metric in km.

Recall that we want to find a spine with high availability measures, and these measures vary on the different spines based on their graphical structure as we showed in the previous section. Here, we involve heterogenous link availability which complicates the problem furthermore. Now, we want to examine to what extent the added input changes the results. To inspect this, we study the Polska and NSF networks, and we calculate distance-based link availability by setting $MTTR = 24$, $CC = 450$, and $a_t = 0.9995$, and using the straight line distance between end-nodes as the cable length. Then, for all STs we calculate A_S and $\text{min-}A_f$ with different Δ values. We compare these results to the results obtained from a homogenous spine case with $a = 0.99$ and $\Delta = 9e - 3$. As an example, Figures 5.15 and 5.16 show a scatterplot for the A_S and $\text{min-}A_f$ values of the STs for Polska and NSF networks. The x -axis and y -axis depict a ST availability value in the heterogenous and homogenous case, respectively. For both A_S and $\text{min-}A_f$, we can see some variation for STs values around the linear correlation line which indicates an STs change in ranking.

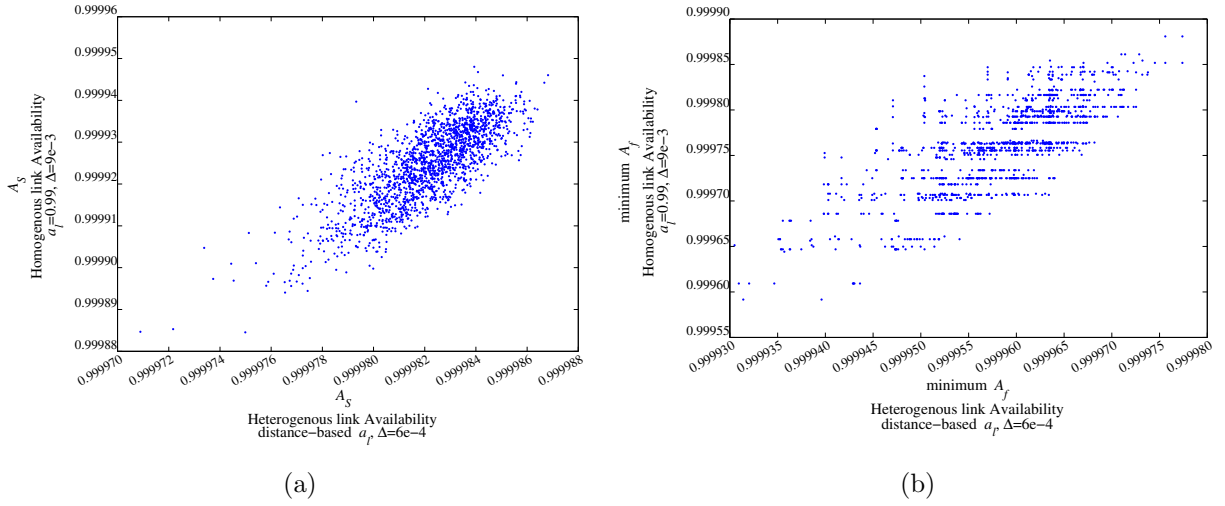


Figure 5.15: Scatterplot of Polska STs measures with homogenous versus heterogenous link availability.

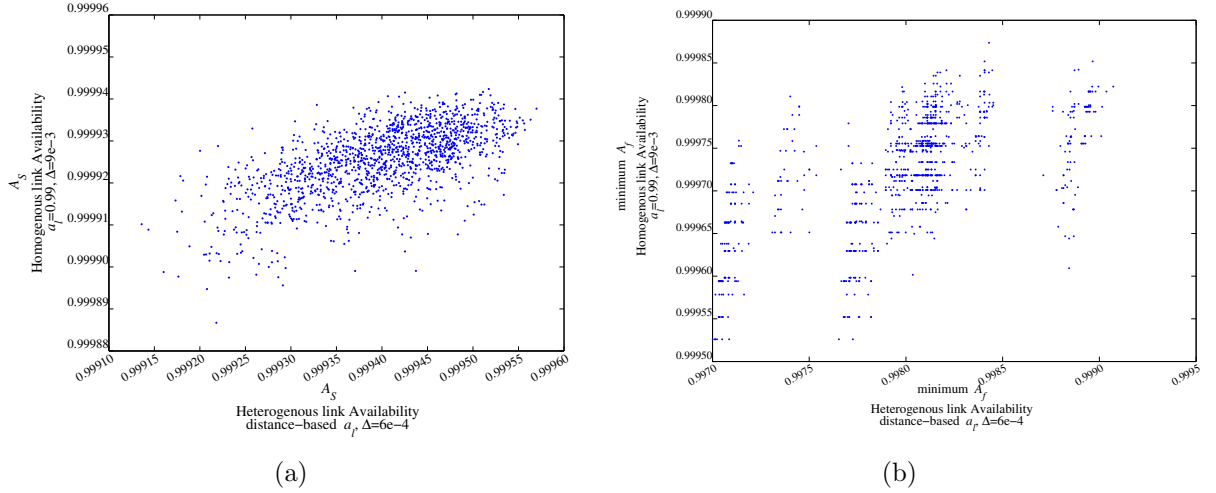


Figure 5.16: Scatterplot of NSF STs measures with homogenous versus heterogenous link availability.

5.4.4 Monetary Cost and Implementation Issues

The discussion and analysis thus far illustrates the potential of the spine concept in improving A_S and reducing the average downtime per year. In reality there are several factors that

will determine the usefulness and practicality of the spine approach. The paramount factor is the financial cost versus benefit tradeoff of the spine approach versus non-spine based methods of improving the availability. Note, that the cost of the spine design is the cost of improving the availability of only the spine components. On the other hand, the cost of the non-spine design is the cost of improving all the components in the network to meet the same average flow availability A_S achieved by the spine design. Hence, the spine is a monetary cost effective option if and only if the cost of the spine design is lower than the cost of the improved non-spine design. This will depend on the financial cost structure of improving the availability for the network under consideration and the desired levels of availability.

In the networking literature the cost of improving availability has not been widely discussed, the majority of papers focus on technical techniques to improve or quantify the availability of components or systems. Financial cost is usually given in a qualitative fashion (*e.g.*, low, medium, high) or in a few cases as a numerical value for a specific technology and application scenario [47]. Determining a precise generally applicable formula on cost of availability is difficult as the cost is dependent on a number of technical and non-technical issues and is typically scenario and organization dependent. In general one can note that the availability of information and communications technology can be improved up to certain point then there are diminishing returns with increased cost and perfect availability is not attainable (*i.e.*, downtime = 0) [121]. A few attempts to provide mathematical models relating cost and availability have appeared in the literature. Grover and Sack [46] proposed to model the reduction of the mean-time-to-repair (MTTR) and the associated cost in terms of % of budget for improving availability as having an inverse relationship of the form $Cost = (MTTR_o/MTTR)^{1/\alpha}$ where $MTTR_o$ is the baseline mean-time-to-repair and α is a parameter. Recently, Herker et al. [122] model the cost of increasing the mean-time-between-failure (MTBF) as polynomial function of MTBF, namely $Cost = MTBF^\alpha + K$ where α is a parameter and K is a constant fixed cost. Note, that these two works each focus on only one side of the techniques to improve availability. In practice, one typically adopts a two-pronged approach to increase availability by investment in organization improvement to reduce MTTR and technical improvements to directly or indirectly (*e.g.*, backup electrical power) increase MTBF [121]. Franke [123] takes a different viewpoint and relates the cost

of improving availability to potential financial loss L due to downtime. The cost function $Netcost = 1 - f(A_o, c)L + c$ is proposed where A_o is the baseline availability, c is the cost of investment in improving the availability as a percentage of L and $f(A_o, c)$ is a nonlinear function relating the improvement in availability as a function of investment. Different forms of $f(A_o, c)$ are proposed such as $f_1(A_o, c) = 1 - (1 - A_o)e^{-\alpha c}$ and $f_2(A_o, c) = 1 - (1 - A_o)/(1 - \alpha c)$.

Another practical deployment issue is that in terms of equipment improvement one does not get continuous changes in the availability but discrete changes in the MTBF by direct component modifications (*e.g.*, spare mirrored line card) or indirect modifications (*e.g.*, backup power supply of 8 hours). However, adjustment of MTTR can occur in a more fine grained fashion. To illustrate this with an example, consider the candidate spines in Table 5.3 in the second and the seventeenth rows which correspond to the Polska and Italia networks, respectively. We compare the total cost of the spine and non-spine designs, where the total cost of a design is the sum of the costs of improving each link. Assume that the availability of all links on the network initially is 0.99, with $MTTR = 24 \text{ hrs}$, which corresponds to $MTBF = 2400$. If we were to increase the MTBF in one link (*i.e.*, to reach a_S or a'_e), the improvement is subject to the cost function $Cost = MTBF^\alpha$ from [122]¹, where the MTBF value is for 1 km. For α , we use different values in the range from 1 to 2 with step size of 0.1, and also we use the geographic distances for links in both networks to calculate the MTBF. In addition, the links on the spine can be given MTTR values from 24 down to 6 hrs, whereas MTTR for the off-spine links in the spine design and all the links in the non-spine design remains unchanged. Figure 5.17 shows the results for the two networks and the two design options. It is clear from the figure that the spine design can be more cost effective for the right combination of both MTBF and MTTR even at large value of α . In practice one will typically not be able to tune the availability in a continuous fashion as in the analysis of previous sections, but there will be discrete options around which the availability can be tuned somewhat as in Figure 5.17.

Lastly we observe that throughout this work we have focused on uncapacitated networks (*i.e.*, not considering links and nodes capacities nor the volume of traffic demands). Since the

¹Note that the cost function here seems to be inaccurate. If one decouples $MTBF$ into $MTTF + MTTR$, link availability can be improved by reducing MTTR but the cost decreases. Thus it is more convenient to have the cost function regarded as $Cost = MTTF^\alpha$.

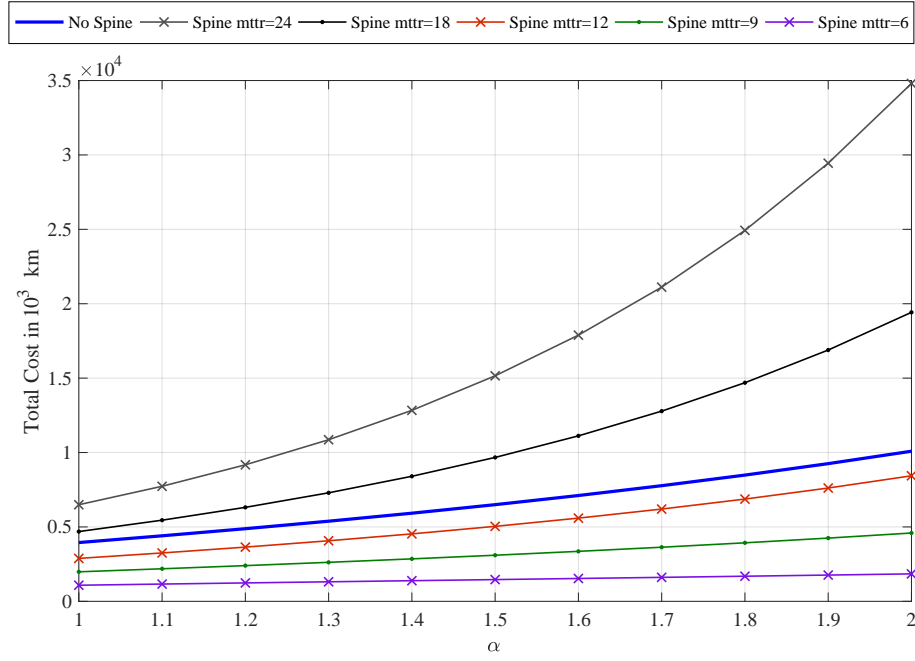
spine can lead to non-shortest path routes it may require more capacity in contrast to a non-spine based design. However, as discussed in the introduction the spine is primarily proposed to satisfy the requirements of high availability traffic and enable the use of QoR classes. Note that only a small fraction of the total flows are expected in the highest availability class, thus one would expect the potential capacity increases to be minimal. In the event the percentage of high availability traffic increases significantly then the capacity of the spine may need to be increased accordingly.

5.5 THE SPINE LINK SELECTION DESIGN PROBLEM

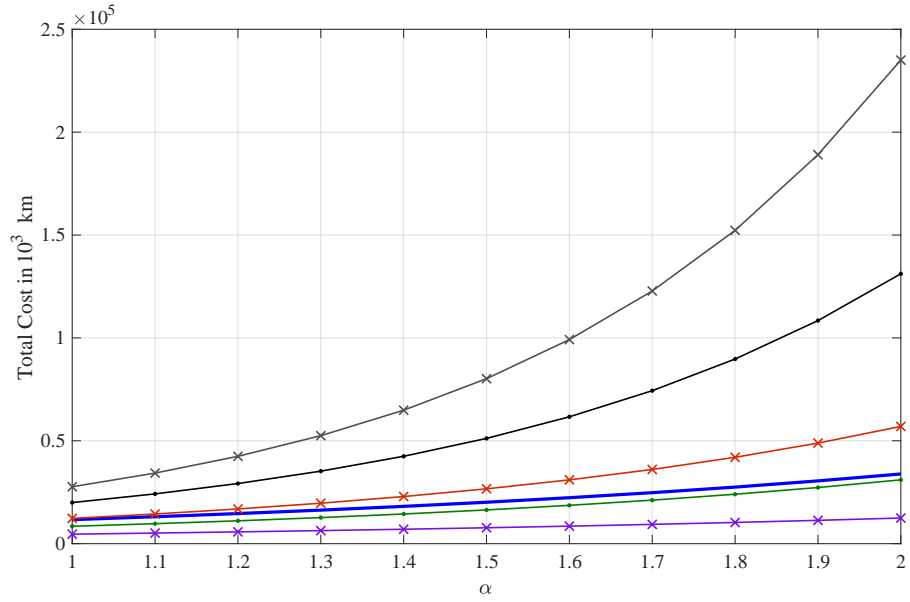
5.5.1 Problem Statement

In Section 5.3, we assumed all links on the spine have the same availability (a_S) and the same is for the links off the spine (a_O). Our sensitivity analysis shows that modifying either value, the improvement step of availability Δ , or considering heterogenous link availabilities would result in a slight change in the ranking of the best spines with respect to the considered availability metrics. Unlike the previous structural-based approach, here we adopt an optimization model approach that involves taking into consideration factors such as the possibility to improve the links and the total budget allocated for improving network resilience. Precisely, the spine structure in addition to the previous requirements comes to be dictated by initial links availability, levels to which links availability can be improved, and the cost associated with these improvements. In general, the goal is to achieve high overall availability for the supported connections of the highest class.

The problem can be restated as follows: Given an optical layer network $\mathcal{G}_P = (\mathcal{V}_P, \mathcal{E}_P)$ with a set of nodes \mathcal{V} and a set of physical links \mathcal{E} , and given a set of the supported end-to-end connections (lightpaths) that need high availability $(s, t) \in \mathcal{F}$, the spine design problem is to find the set of physical links, \mathcal{E} , that forms the spine $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$, where $\mathcal{G}_S \subset \mathcal{G}$. We assume a cost c_{ij}^k associated with each link that is the cost of improving link (i, j) using method k , and the spine formation is constrained by a total budget C .



(a) spine in the second row of Table 5.3



(b) spine in the seventeenth row of Table 5.3

Figure 5.17: Comparison between total cost of the spine design and non-spine design.

5.5.2 Notation

In addition to the notation given in Section 5.3, we define the following:

Indices:

ij	represent a physical link by its two end-nodes, $i, j \in \mathcal{V}_P$, $(i, j) \in \mathcal{E}_P$.
st	represent a connection/flow between two end-nodes, $s, t \in \mathcal{V}_P$, $(s, t) \in \mathcal{F}$.

Parameters:

δ	a scaling factor ≥ 1 .
k	method of link improvement.
a_{ij}	initial link availability.
a_{ij}^k	availability of link ij after applying improvement option k , with $a_{ij}^1 = a_{ij}$.
$\widehat{a_{wp}}(\widehat{a_{bp}})$	flow WP (BP) availability target.
c_{ij}^k	cost of improving link (i, j) using method k .
C	total cost (budget).

Variables:

$x_{ij}^{st} (y_{ij}^{st})$	a binary variable denoting whether physical link (i, j) is used for routing the WP (BP) of connection (s, t) .
x_{ij}	a binary variable indicating whether link (i, j) is selected on the spine ($x_{ij}=1$) or not ($x_{ij}=0$).
r_{ij}^k	a binary variable indicating whether method k is used for link (i, j) .
$p_{ij}^{st} (q_{ij}^{st})$	a continuous variable denoting link (i, j) unavailability given that it is on connection (s, t) working path (backup path).

5.5.3 Incremental Link Availability Model

Network operators continually collect and analyze failure logs so that accurate equipment and link availability values are obtainable. In addition, an appropriate effort is expected to be placed in improving equipment availability and properly managing maintenance and repair duties in a way so that new improved availability figures are predictable.

Hence consider that each link in a given network, $e \in \mathcal{E}$, can be purposely strengthened so that its MTBF is increased, for example by altering the cable implementation method (*e.g.*, burying an aerial cable) or adding physical protection or the MTTR is reduced by intensive maintenance and repair efforts as discussed in Section 2.2.4. For each link, one can collect possible options to improve its availability, and each option would result in a different availability level and incurred cost. Specifically, if the link e spans node-pair ij and has availability a_{ij} , using method k , the link availability can be improved to a_{ij}^k with cost c_{ij}^k , whereas using method $k + 1$ that costs c_{ij}^{k+1} , availability is improved to a_{ij}^{k+1} .

Due to the difficulty in getting exact availability values and the associated cost of improvement from network operators, as discussed in Section 2.2.3, we use synthetic availability levels and improvement costs to conduct our study. Most of the assumptions made here can be justified by taking into account the characteristics of various physical layer cable technology and the maintenance/repair techniques adopted.

For a given network, each link is assigned an initial link availability value a_{ij} based on its length, with longer links being less reliable. The link availability values are within $(\check{a} - \hat{a})$, where \hat{a} is the maximum link availability assigned to the shortest link and \check{a} is the minimum link availability assigned to the longest link. Then, for each link we assume “ K ” possible availability values $(a_{ij}^k, k = 1, 2, \dots, K)$ with $(a_{ij}^1 = a_{ij})$. For each value k , the corresponding unavailability is reduced by ϵ %, so that $u_{ij}^k = u_{ij}^{k-1} \cdot (1 - \epsilon)$, where $u_{ij}^k = 1 - a_{ij}^k$. Reducing a link unavailability is analogous to reducing its expected downtime. It can also be expressed in terms of availability as $a_{ij}^k = a_{ij}^{k-1}(1 - \epsilon) + \epsilon$. Note that, in reality the different options k might not have fixed downtime differences within nor across links. For example, it may turn out that for a given link, the availability differences of two or three improvement options are small. Here, we choose a fixed ϵ for illustration purpose. In practice, not all links will likely have the same number of options as this depends on several factors (*e.g.*, the terrain, cable type, the associated cost, etc). However, we assume that this is the case here in order to simplify the model. The cost associated with each improvement step k is calculated by a given cost function, $c_{ij}^k = f_c(a_{ij}^k, a_{ij}^1)$.

Figure 5.18 shows the Polska network topology and a sample of the availability options for three different links with $K = 7$. Each table in the figure shows the availability levels of

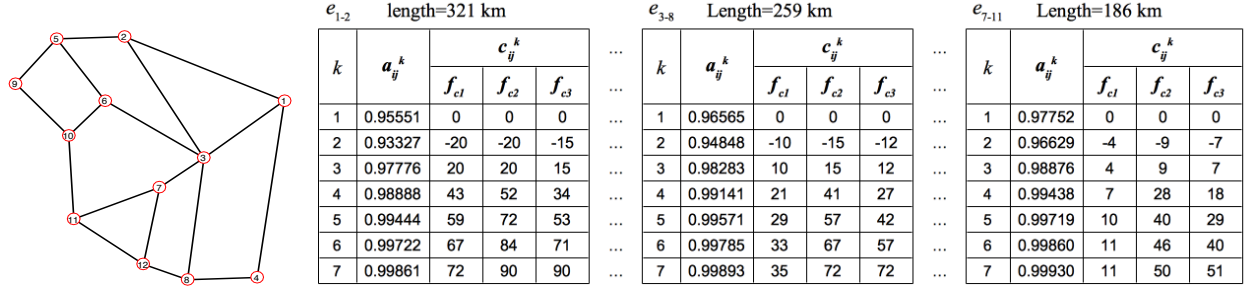


Figure 5.18: An example of the incremental link availability model for the Polska network.

a link and the corresponding cost. In the figure, $k = 1$ denotes the initial link availability. The case of $k = 2$ is set to model transferring maintenance capabilities between links. Thus the expected downtime of some links (*i.e.*, off the spine) would increase and incur negative cost which would give an extra allowance to the total budget C . Subsequently, other links availability (*i.e.*, on the spine) can be improved by the relocation of maintenance and repair capabilities, and take advantage of transferring operational expenditure from degraded links. Therefore, we set $u_{ij}^2 = u_{ij}^1 \cdot (1 + \epsilon)$ and $c_{ij}^2 = -c_{ij}^3$.

5.5.4 Optimization Model Formulations

The spine design problem aims at finding the best combination of links to form the spine and the improvement options for all links in order to achieve better overall availability and widen the range of availabilities. First, we need to route all lightpaths on the spine with fully link-disjoint backup paths. This ensures that all high priority traffic supported by the spine can be given a 1+1 dedicated protection. Note that, this also enables 1:N shared protection, however this topic is left for future work. Second, we assume that the class of critical services is required between all possible node-pairs *i.e.*, a full mesh of demand of one unit between each node pair. To consider a spine with full connectivity, the spine structure can be a minimum spanning tree (MST) similar to the model in Section 5.3. Thirdly, flow availability is constrained to be greater or equal to target values $(\widehat{a}_{wp}, \widehat{a}_{bp})$. Instead of looking for the spine that maximizes the average availability, here we require that

the minimum WP availability on the spine to be within an acceptable range of availability, $\widehat{a_{wp}}$. As showed earlier in Section 5.1, having a large availability on WP improves the end to end availability more effectively. In addition, the design has to constrain the expected increase in total resources for routing working and backup paths for all connections. Finally, the objective \mathcal{C} of the design problem aims at minimizing the total costs of embedding the spine and improving flows availabilities. Given the notation above, the spine link selection optimization problem can be formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & C = \sum_{ij} \sum_k r_{ij}^k \times c_{ij}^k \\ \text{s.t.} \end{aligned} \quad (5.17)$$

WP and BP computation:

$$\sum_{hj \in \mathcal{E}_P} x_{hj}^{st} - \sum_{ih \in \mathcal{E}_P} x_{ih}^{st} = \begin{cases} 1 & \text{if } h = s \\ -1 & \text{if } h = t \\ 0 & \text{otherwise} \end{cases}, \forall h \in \mathcal{V}_P, (s, t) \in \mathcal{F} \quad (5.18)$$

$$\sum_{hj \in \mathcal{E}_P} y_{hj}^{st} - \sum_{ih \in \mathcal{E}_P} y_{ih}^{st} = \begin{cases} 1 & \text{if } h = s \\ -1 & \text{if } h = t \\ 0 & \text{otherwise} \end{cases}, \forall h \in \mathcal{V}_P, (s, t) \in \mathcal{F} \quad (5.19)$$

Loopless Routing:

$$x_{ij}^{st} + x_{ji}^{st} \leq 1, \forall (i, j) \in \mathcal{E}_P, (s, t) \in \mathcal{F} \quad (5.20)$$

$$y_{ij}^{st} + y_{ji}^{st} \leq 1$$

$$\sum_{\substack{j \in \mathcal{V}_P \\ ij \in \mathcal{E}_P}} x_{ij}^{st} + \sum_{\substack{h \in \mathcal{V}_P \\ hi \in \mathcal{E}_P}} x_{hi}^{st} \leq 2, \forall i \in \mathcal{V}_P, (s, t) \in \mathcal{F} \quad (5.21)$$

$$\sum_{\substack{j \in \mathcal{V}_P \\ ij \in \mathcal{E}_P}} y_{ij}^{st} + \sum_{\substack{h \in \mathcal{V}_P \\ hi \in \mathcal{E}_P}} y_{hi}^{st} \leq 2, \forall i \in \mathcal{V}_P, (s, t) \in \mathcal{F} \quad (5.22)$$

Disjointness constraints:

$$x_{ij}^{st} + y_{ij}^{st} \leq 1 \quad x_{ij}^{st} + y_{ji}^{st} \leq 1, \forall (i, j) \in \mathcal{E}_P, (s, t) \in \mathcal{F} \quad (5.23)$$

Hop-count constraint:

$$H_S = \sum_{ij \in \mathcal{E}_P} \sum_{st \in \mathcal{F}} (x_{ij}^{st} + y_{ij}^{st}) \leq \delta \times H_G \quad (5.24)$$

MST formation:

$$x_{ij} \geq x_{ij}^{st}, \quad \forall (s, t) \in \mathcal{F} \quad (5.25)$$

$$\sum_{ij \in \mathcal{E}_P} x_{ij} \leq |\mathcal{V}_P| - 1 \quad (5.26)$$

Availability constraints:

$$\sum_k r_{ij}^k = 1, \quad \forall (i, j) \in \mathcal{E}_P \quad (5.27)$$

$$r_{ij}^k = r_{ji}^k, \quad \forall (i, j) \in \mathcal{E}_P \quad (5.28)$$

$$p_{ij}^{st} = x_{ij}^{st} \times \sum_k r_{ij}^k (1 - a_{ij}^k), \quad \forall (i, j) \in \mathcal{E}_P, (s, t) \in \mathcal{F} \quad (5.29)$$

$$q_{ij}^{st} = y_{ij}^{st} \times \sum_k r_{ij}^k (1 - a_{ij}^k), \quad \forall (i, j) \in \mathcal{E}_P, (s, t) \in \mathcal{F} \quad (5.30)$$

Flow availability targets:

$$A_{st}^{WP} = 1 - \sum_{ij \in \mathcal{E}_P} p_{ij}^{st} \geq \widehat{a}_{wp} \quad (5.31)$$

$$A_{st}^{BP} = 1 - \sum_{ij \in \mathcal{E}_P} q_{ij}^{st} \geq \widehat{a}_{bp} \quad (5.32)$$

Variables:

$$x_{ij}^{st}, y_{ij}^{st}, x_{ij}, r_{ij}^k \quad \text{binary} \quad (5.33)$$

$$p_{ij}^{st}, q_{ij}^{st} \in [0, 1] \quad (5.34)$$

We consider a variation of the classical network flow problem. In the core of the formulation is the flow conservation constraints. Constraint sets (5.18) and (5.19) find primary and backup paths for all lightpaths, respectively. A flow conservation constraint pushes a unit of flow along a path between the two end-nodes of a given lightpath. Constraint sets (5.20)-(5.22) ensure loop free routing. Constraint set (5.23) ensures that for each lightpath

the primary and backup paths are fully link-disjoint. The sum of these paths, however, is constrained by (5.24) which sets a maximum limit H_G for the hop count of the link-disjoint path-pairs and δ is a scaling factor. Each link used by a primary path of any lightpath is considered as an on-spine link. Constraint (5.25) enforces this by turning the spine link selector variable for a link, x_{ij} , to 1 if the link is used in a primary path of at least one lightpath. Then, constraint (5.26) limits the number of the links selected for the spine to $|\mathcal{V}_P| - 1$ which is the number of links for a MST. Next, for the availability constraints, constraint set (5.27) ensures that only one improvement method is selected for each link. Constraint (5.28) requires that a link has the same improvement method in both directions. Constraint sets (5.29) and (5.30) are used to relate a flow WP and BP unavailability to the unavailability of each link along the flow path. Variable p_{ij}^{st} or q_{ij}^{st} will have an unavailability value only if flow (s, t) WP or BP is routed through link (i, j) . These two sets of constraints, turn the optimization problem into a integer nonlinear programming (INLP), because the product of two variables *i.e.*, x_{ij}^{st} with r_{ij}^k in (5.29) and y_{ij}^{st} with r_{ij}^k in (5.30). Note that, to compute a single path availability for a given flow, one can multiply the availability of the links along the path, but this results in a nonlinearity. Instead, we use the approximate version of the unavailability formula for a system connected in series, ($u^{st} \approx \sum_{ij} u_{ij}^{st}$). Hence, WP availability can be computed as $(1 - \sum_{ij} p_{ij}^{st})$. BP availability is computed in the same way. Constraints (5.31) and (5.32) require that a flow WP and BP are above target availability values $\widehat{a_{wp}}$ and $\widehat{a_{bp}}$, respectively. Lastly, constraint sets (5.33) and (5.34) declare binary and continuous variables.

To remove the nonlinearity of the INLP, constraints set (5.29) can be replaced with constraint sets eqs. (5.35) to (5.37). The three constraints provide the same function as (5.29). Similarly, constraints set (5.30) that computes BP unavailability can be replaced with the set of eqs. (5.38) to (5.40).

Linearized availability constraints:

$$p_{ij}^{st} \leq x_{ij}^{st} \tag{5.35}$$

$$p_{ij}^{st} \leq \sum_k r_{ij}^k (1 - a_{ij}^k) \quad (5.36)$$

$$p_{ij}^{st} \geq x_{ij}^{st} + \sum_k r_{ij}^k (1 - a_{ij}^k) - 1, \forall (i, j) \in \mathcal{E}_P, (s, t) \in \mathcal{F} \quad (5.37)$$

$$q_{ij}^{st} \leq y_{ij}^{st} \quad (5.38)$$

$$q_{ij}^{st} \leq \sum_k r_{ij}^k (1 - a_{ij}^k) \quad (5.39)$$

$$q_{ij}^{st} \geq y_{ij}^{st} + \sum_k r_{ij}^k (1 - a_{ij}^k) - 1, \forall (i, j) \in \mathcal{E}_P, (s, t) \in \mathcal{F} \quad (5.40)$$

Although nonlinearity constraints are avoided, the spine link selection design problem is NP-complete since the optimization version of the problem of finding disjoint path-pair is known to be NP-complete [124–126]. This means that it is difficult to solve the problem for optimality. Instead, we can find a feasible solution within an optimality gap from a solution bound. Typically, this bound is the solution of the relaxed version of the problem, in which all integer variables are processed as continuous. There is a number of well known relaxation and heuristics methods for solving integer programming problems and iteratively trying to minimize the optimality gap. Thus one can expect solving the design problem efficiently for moderate sized networks and the chance of proving optimality becomes strongly size dependent.

Lastly before we move to our numerical analysis, we examine the quality of a homogenous spine obtained from the optimization model and compare it to the spines obtained from heuristics in Sections 5.3 and 5.4, respectively, along with the set of all possible spanning trees. We consider the Polska network with $K = 2$, and two link availability values; $a_{ij}^1 = 0.99$ for off the spine and $a_{ij}^2 = 0.999$ for the on spine links. We solved the optimization problem with a minimum WP availability goal $\widehat{a_{wp}} = 0.995$ and the BP availability goal $\widehat{a_{bp}}$ of constraint (5.32) is relaxed. Then we calculated the average A_S^{WP} , A_S , and $\min-A_f$ for the resulted spine and add them to the corresponding plots of Figure 5.8. Figure 5.19 shows the results for the spine obtained by the ILP as green circles along each line. We can see that the obtained spine has a considerably large score with respect to A_S^{WP} and A_S , and a reasonably large $\min-A_f$. The result reconfirms that by ensuring a minimum flow WP availability on

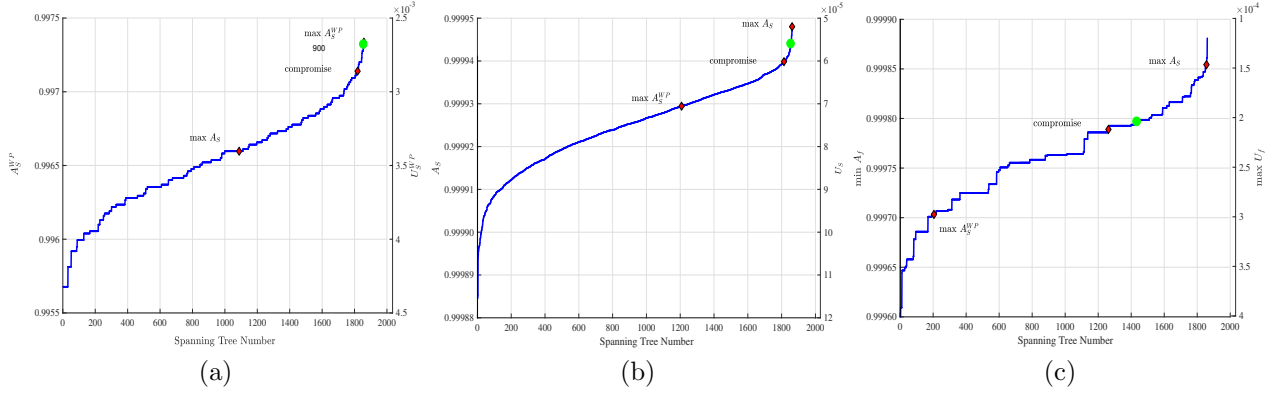


Figure 5.19: The optimal spine with $a_{ij}^1 = 0.99$ and $a_{ij}^2 = 0.999$ (in green) compared to heuristic spines and all spanning trees of the Polska Network.

a spine, we can obtain a fairly good spine with high availability compared to all possible spanning trees in a network.

5.5.5 Numerical Study

We consider the Polska, Spain, and Italia14 networks shown in Table 5.2 to evaluate our model. The table shows the number of nodes and links, the density ratio, and the diameter of each network. We set $\tilde{a} = 0.95$, $\hat{a} = 0.995$, $\epsilon = 0.50$, $K = 7$, and δ in constraint (5.24) is set to 110%, allowing for a maximum of 10% increase in total resources over the resources required by shortest path-pairs, H_G (*i.e.*, $H_S \leq 1.1H_G$).

5.5.5.1 Cost Functions As mentioned earlier, a precise formula for cost of availability is unobtainable. Instead, many researcher rely on some mathematically known models (*e.g.*, constant, linear, quadratic, exponential, etc) to relate cost to availability [127]. Typically, more than one model is involved in a single study to account for the imprecision and offer a range of prospects about the subject of the study. Here, we consider that the cost of improving a link becomes larger as the availability gets higher [8], and this can take several formats. For example, the larger the difference between the initial availability and improved availability, the higher the cost, or improving an already high availability link costs more than

improving a link with a moderate availability. A third format might feature different scales for each range of availability values. To capture these formats, we consider the following cost functions, f_c 's, to compute the cost of improving the link availability per unit of length:

- The first cost function, f_{c1} , is a polynomial in the availability improvement $\Delta a_{ij}^k = a_{ij}^k - a_{ij}^1$.

$$f_{c1}(a_{ij}^k, a_{ij}^1) = (a_{ij}^k - a_{ij}^1)^\alpha, \quad k > 2 \quad (5.41)$$

where α is a scaling parameter. This function represents the first format.

- The second cost function, f_{c2} , represents the second format and it is a polynomial in the availability improvement $\Delta a_{ij}^k = a_{ij}^k - a_{ij}^1$ but also weighted by the unavailability of the link. Hence for equal Δa_{ij}^k , it compounds the cost for the link with higher availability. This formula is very similar to f_2 in [123] but with different application of the exponent α where it is applied as a divisor scaling parameter.

$$f_{c2}(a_{ij}^k, a_{ij}^1) = \left(\frac{a_{ij}^k - a_{ij}^1}{1 - a_{ij}^1} \right)^\alpha, \quad k > 2 \quad (5.42)$$

- The third cost function, f_{c3} , is derived from f_1 in [123], shown in section 5.4.4, where $a_{ij}^k = 1 - (1 - a_{ij}^1)e^{-\alpha a_{ij}^k}$, meaning that the impact of the cost on the improved availability decreases exponentially.

$$f_{c3}(a_{ij}^k, a_{ij}^1) = -\ln \left[\frac{1 - a_{ij}^k}{1 - a_{ij}^1} \right], \quad k > 2 \quad (5.43)$$

In addition, to include the length factor in the upgrading cost we let,

$$c_{ij}^k = f_c(a_{ij}^k, a_{ij}^1) \times d_{ij} \quad (5.44)$$

where d_{ij} is link (i, j) length. The exponent α in (5.41) and (5.42) was set to 2 to impose quadratic growth of the cost. Figure 5.20 shows the link availability and the corresponding cost using the three cost functions for the three links shown in Figure 5.18. Observe that cost function f_{c1} has a smaller range of values than the other two cost functions and that the shorter the link the less the cost to improve the availability. In addition, Figure 5.21 plots the CDF of the cost values for all links in the Polska, Spain, and Italia14 networks. Note that the cost values within each cost function are normalized and scaled between 1~100.

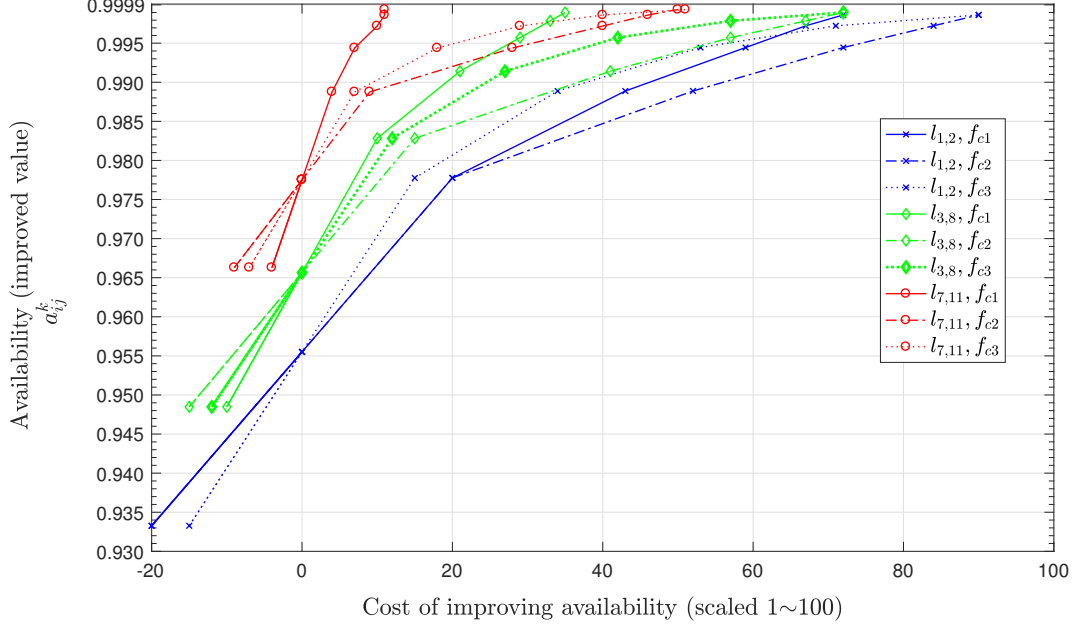


Figure 5.20: Improved link availability versus cost for three links from the Polska network shown in Figure 5.18.

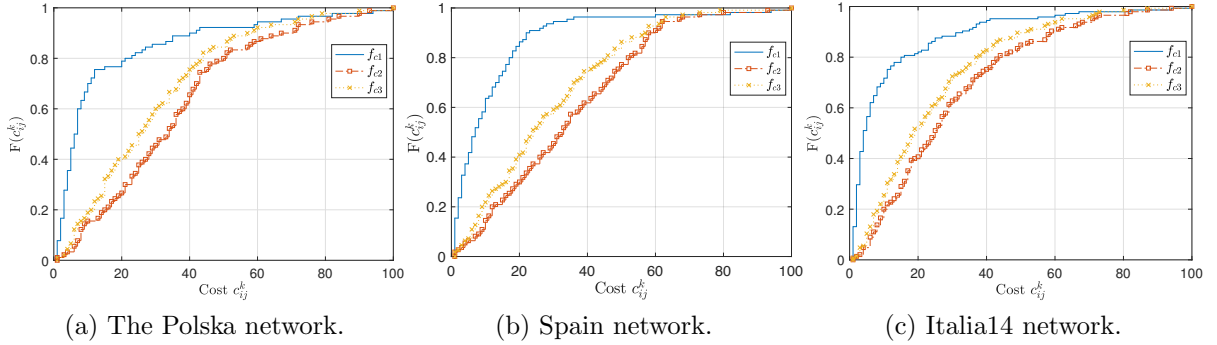


Figure 5.21: CDF of the link improvement costs.

One can see that for f_{c1} , 75% of the cost values are below 20 for the Polska network whereas only 27% and 40% of these values are below 20 for f_{c2} and f_{c3} , respectively. A comparable behavior can be seen for the other network.

5.5.5.2 Results We solved the spine design optimization problem for the test networks using AMPL/Gurobi and present a sample of the results. Here, the availability goal for WP, \widehat{a}_{wp} in Equation (5.31), is set 0.99 and then increased in steps until the model becomes infeasible. At each step, we solve the model for the three cost functions. We ignore \widehat{a}_{bp} for now. For the Polska network $\widehat{a}_{wp} = \{0.99, 0.995, 0.996, 0.9964\}$, for the Spain network $\widehat{a}_{wp} = \{0.99, 0.996, 0.9964, 0.9967\}$, and for the Italia14 network $\widehat{a}_{wp} = \{0.99, 0.995, 0.996, 0.997\}$. Increasing the target value \widehat{a}_{wp} increases the total cost of the design and may also result in a different spine layout. Figures 5.22, 5.25 and 5.28 show the spine layouts which resulted for the Polska, Spain, and Italia14 networks, respectively, as the target \widehat{a}_{wp} increases. First, the spine layout varies slightly as the target availability, \widehat{a}_{wp} , or the cost function changes. Though one can see that in each network there is a persistence substructure that appears in almost all the spines, *e.g.*, the star-like substructure rooted at node 3 in the Polska network. In general, the spines tend to have a star-like layout as the diameters of these spines are considered short as shown in Table 5.6. In addition, Table 5.6 shows that the spines tend to present comparatively a small edge betweenness eb_S and average shortest path h_S and a large edge degree ed_S . Only in a few cases does the corresponding measure match the minimum (or maximum) value across all MSTs shown in Table 5.5. These results comply with the findings in Section 5.3 with respect to the spine that maximizes A_S^{WP} and indicate that the spines tend to have a star-like rather than a ring-like layout. This behavior can also be traced in Figures 5.24, 5.27 and 5.30 that plot node degrees in the full network and in the spines *e.g.*, for the Italia14 network, either node 7 or 11 maintained a high degree across scenarios and at least 7 nodes are leafs. Overall, there is a similar pattern of the node degrees within and across the different cost functions. Moreover, as the spine layout might be attributed to the structural importance of the links and nodes, it is also shaped by the cost associated with the links and their availability as well as the hop-count constraint. This justifies the small variations in the structural properties between the spines obtained here and in Table 5.5. Similar spines within and across cost functions are likely to have different availability and link types. This is illustrated by Figures 5.23, 5.26 and 5.29 where downtime per year for each link versus the link length and the link improvement method/type k selected for each link is shown for the cases corresponding to Figures 5.22, 5.25 and 5.28. In the figures, each

Table 5.6: Structural properties of the resulted spines.

(a) Polska.						(b) Spain.						(c) Italia14.					
cost function	scenario	eb_S	ed_S	h_S	di_S	cost function	scenario	eb_S	ed_S	h_S	di_S	cost function	scenario	eb_S	ed_S	h_S	di_S
	\mathcal{G}_g	0.12	6.33	2.14	4		\mathcal{G}_g	0.1	6.82	2.27	5		\mathcal{G}_g	0.06	10.3	1.87	3
f_{c1}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.26	4.91	2.89	6	f_{c1}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.26	4.46	3.38	7	f_{c1}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.23	5.38	2.97	6
	$\mathcal{G}_S, \widehat{a_{wp2}}$	0.26	4.91	2.89	6		$\mathcal{G}_S, \widehat{a_{wp2}}$	0.26	4.46	3.38	7		$\mathcal{G}_S, \widehat{a_{wp2}}$	0.23	5.38	2.97	6
	$\mathcal{G}_S, \widehat{a_{wp3}}$	0.26	4.91	2.89	6		$\mathcal{G}_S, \widehat{a_{wp3}}$	0.26	4.46	3.38	7		$\mathcal{G}_S, \widehat{a_{wp3}}$	0.22	5.54	2.86	6
	$\mathcal{G}_S, \widehat{a_{wp4}}$	0.24	5.45	2.68	5		$\mathcal{G}_S, \widehat{a_{wp4}}$	0.24	4.62	3.11	6		$\mathcal{G}_S, \widehat{a_{wp4}}$	0.21	5.85	2.79	6
f_{c2}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.26	4.91	2.89	6	f_{c2}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.25	4.62	3.21	7	f_{c2}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.21	6.00	2.71	5
	$\mathcal{G}_S, \widehat{a_{wp2}}$	0.26	4.91	2.89	6		$\mathcal{G}_S, \widehat{a_{wp2}}$	0.26	4.46	3.38	7		$\mathcal{G}_S, \widehat{a_{wp2}}$	0.21	6.00	2.71	5
	$\mathcal{G}_S, \widehat{a_{wp3}}$	0.26	4.91	2.89	6		$\mathcal{G}_S, \widehat{a_{wp3}}$	0.26	4.46	3.38	7		$\mathcal{G}_S, \widehat{a_{wp3}}$	0.21	6.00	2.71	5
	$\mathcal{G}_S, \widehat{a_{wp4}}$	0.24	5.45	2.68	5		$\mathcal{G}_S, \widehat{a_{wp4}}$	0.24	4.62	3.11	6		$\mathcal{G}_S, \widehat{a_{wp4}}$	0.21	5.85	2.79	6
f_{c3}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.26	4.91	2.82	5	f_{c3}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.25	4.62	3.21	7	f_{c3}	$\mathcal{G}_S, \widehat{a_{wp1}}$	0.21	6.00	2.71	5
	$\mathcal{G}_S, \widehat{a_{wp2}}$	0.26	4.91	2.82	5		$\mathcal{G}_S, \widehat{a_{wp2}}$	0.26	4.46	3.38	7		$\mathcal{G}_S, \widehat{a_{wp2}}$	0.21	6.00	2.71	5
	$\mathcal{G}_S, \widehat{a_{wp3}}$	0.26	4.91	2.89	6		$\mathcal{G}_S, \widehat{a_{wp3}}$	0.26	4.46	3.38	7		$\mathcal{G}_S, \widehat{a_{wp3}}$	0.21	6.00	2.71	5
	$\mathcal{G}_S, \widehat{a_{wp4}}$	0.24	5.45	2.68	5		$\mathcal{G}_S, \widehat{a_{wp4}}$	0.24	4.62	3.11	6		$\mathcal{G}_S, \widehat{a_{wp4}}$	0.21	6.31	2.70	5

circle represents a link and the number inside the circle is the improvement method/type k . The red circles represent links comprising the spine and the blue are the off spine links. For example, the spines obtained for the Polska network with $\widehat{a_{wp4}}$ for cost functions f_{c1} and f_{c2} are identical in the layout, as shown in Figures 5.22d and 5.22h, but the corresponding link improvement method k of the links are different as shown in Figures 5.23d and 5.23h. However, the spine and the selected methods are identical for f_{c2} and f_{c3} , whereas, they were completely different for $\widehat{a_{wp1}}$ as shown in Figures 5.22e and 5.22i. One also can see that, within the same cost function scenarios, different methods k can be selected as the $\widehat{a_{wp}}$ changes. For example, the first three spine obtained for the Polska network for cost function f_{c1} , shown in Figures 5.22a to 5.22c, have different link improvement assignments as WP availability target $\widehat{a_{wp}}$ changes. This is shown in the downtime and availability assignment Figures 5.23a to 5.23c. Initially, shorter links (*i.e.*, with higher availability and lower improvement cost) are favored to be selected as a spine link, thus exploiting existing heterogeneity. Then for some high availability target, expensive links may be selected due to structural motives. That is to achieve $\widehat{a_{wp4}}$ in the previous example, the spine layout changes from the initial one in order to achieve this target. Structurally, we noticed that link (2,3) (that is, the third longest link) is selected to be on the spine despite. Also Table 5.6 shows that the obtained spine have better structural measures. The results for Spain and Italia14 networks exhibit similar observations. Notice that the off spine links are selected as type

$k = 2$ which are less reliable, but provide additional budget resources to improve the spine links from $k = 1$ to better quality links (*i.e.*, $k > 2$).

Second, we compare the average expected flow downtime A_S and average expected WP downtime A_S^{WP} for the different scenarios and cost functions. We also include the corresponding downtime of an equivalent network with no spine (A_G, A_G^{WP}). In the no spine network, all links are improved using the same method, k , and the total cost C is computed accordingly. Figures 5.31 to 5.33 show the average expected WP and end-to-end flow downtimes for the three test networks considering the different cost functions and $\widehat{a_{wp}}$'s. Also the results are shown for two cases: one that forbids relaxing the MTTR for the off spine links *i.e.*, $k = 1$ (continuous line) and one that permits such relaxation *i.e.*, $k = 2$ (dot-dashed line). From the figures we can make a number of observations. For the Polska and Spain networks, there is significant improvement in the downtime values over the no spine model for the case of cost function f_{c1} , slight improvement for the case of f_{c2} , and no improvement f_{c3} . However, the dense network, Italia14, achieves lower downtime in all cases across all cost functions. These findings correlate the efficiency of the spine model to both the network density and the distribution of the cost of improving links availability. Lastly, comparing relaxable and non-relaxable MTTR cases shows a significant saving in cost when relaxing the off spine links MTTR for a slight decrease in the average flow downtime A_S .

Recall that the spine concept aims to create different levels of availability and also meet the most stringent availability requirements. Figures 5.34 to 5.36 show the expected downtime for each path type for the optimal spines obtained for each network. The downtime results are represented for each scenario as a box plot. The upper and lower edges of each box represent the third and first quartile of the values, respectively, the middle bar (in red) represents the median, and the upper and lower bars represent the maximum and minimum downtime values across all paths, respectively. Note that, even for the spine with the lowest cost (*i.e.*, $\widehat{a_{wp1}}$ and relaxable *MTTR*), there are three different levels of availability classes resulting from using only one protection scheme. The lower availability class can be given an unprotected path equivalent to the backup path with large expected downtime. Then, the middle class is routed on an unprotected path on the spine which achieves shorter expected downtime compared to the lower class. The higher class is routed on the spine and protected

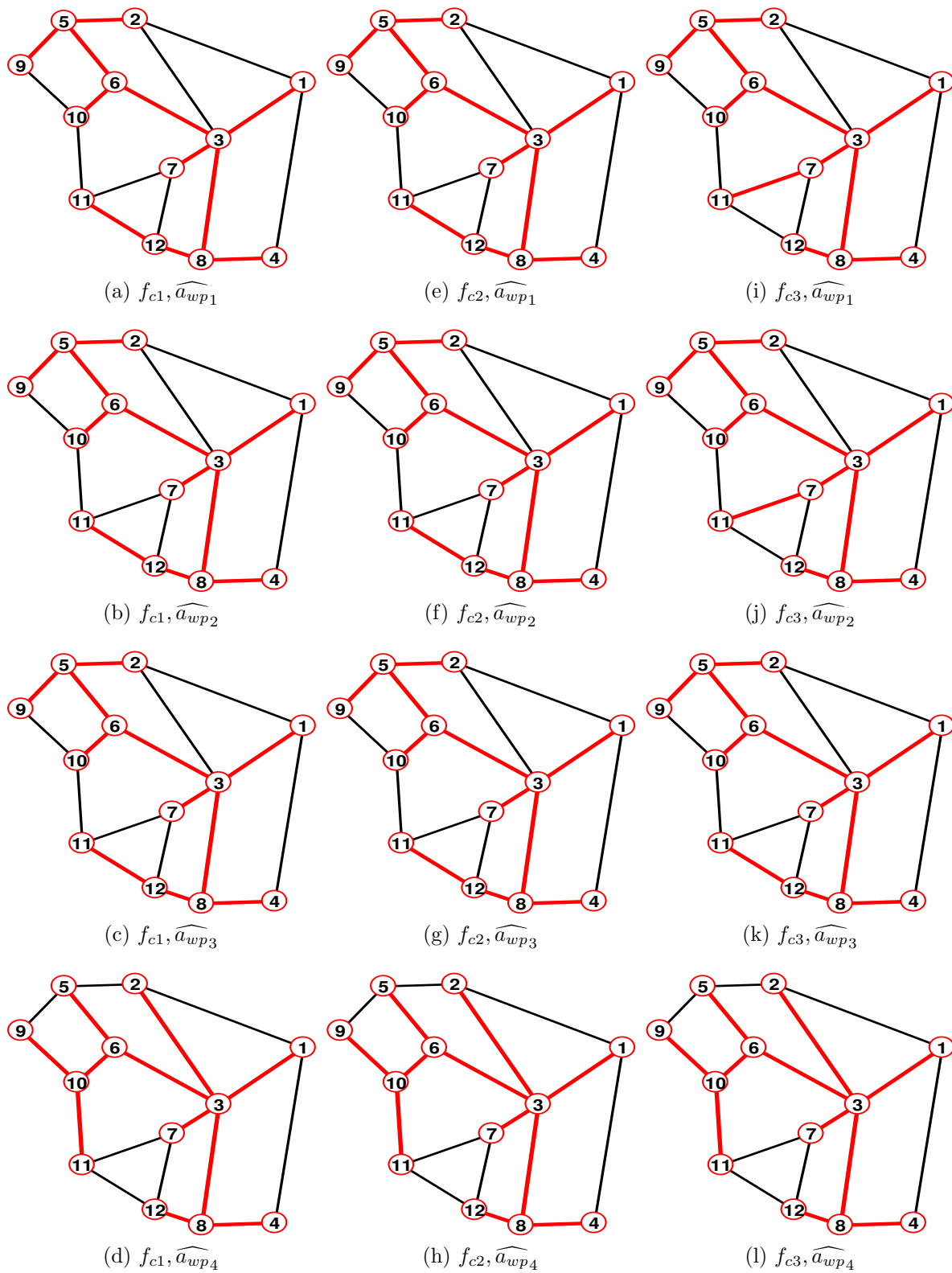


Figure 5.22: Spines obtained for the Polska network using the three cost functions.

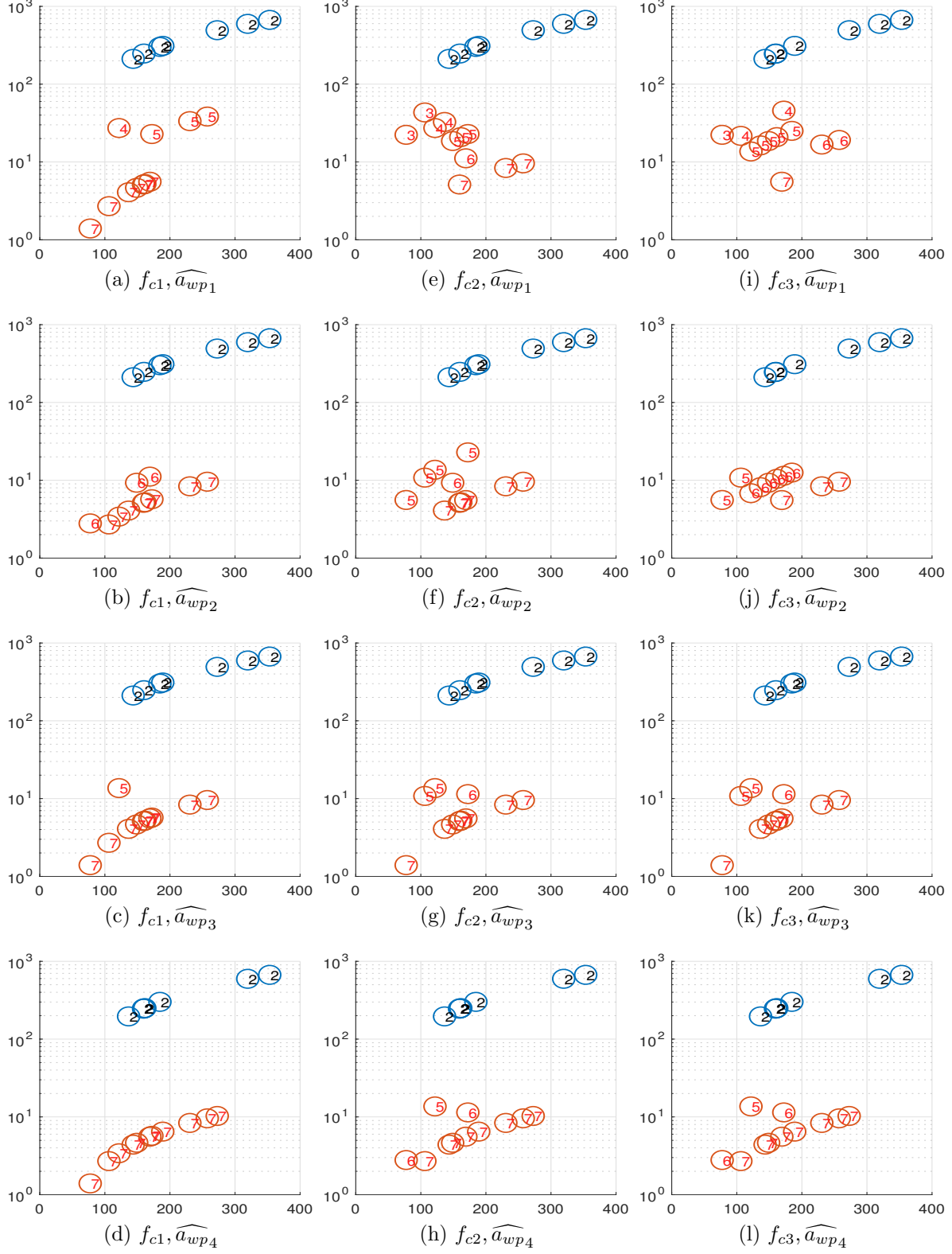
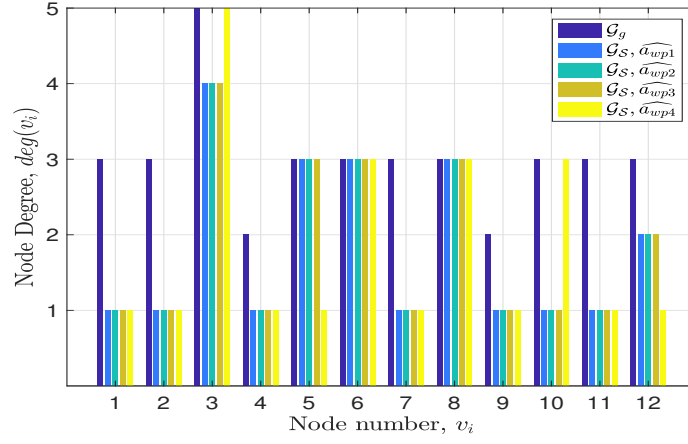
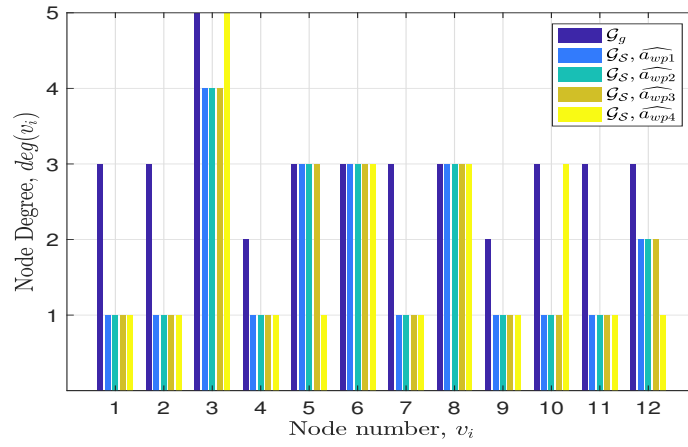


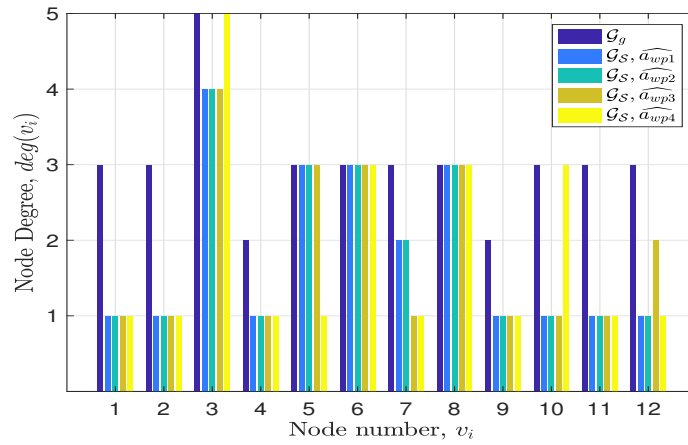
Figure 5.23: The corresponding link downtime/year and versus link length for the spines obtained for the Polska network.



(a) f_{c1} .



(b) f_{c2} .



(c) f_{c3} .

Figure 5.24: Node Degree for the resulted Polska spines.

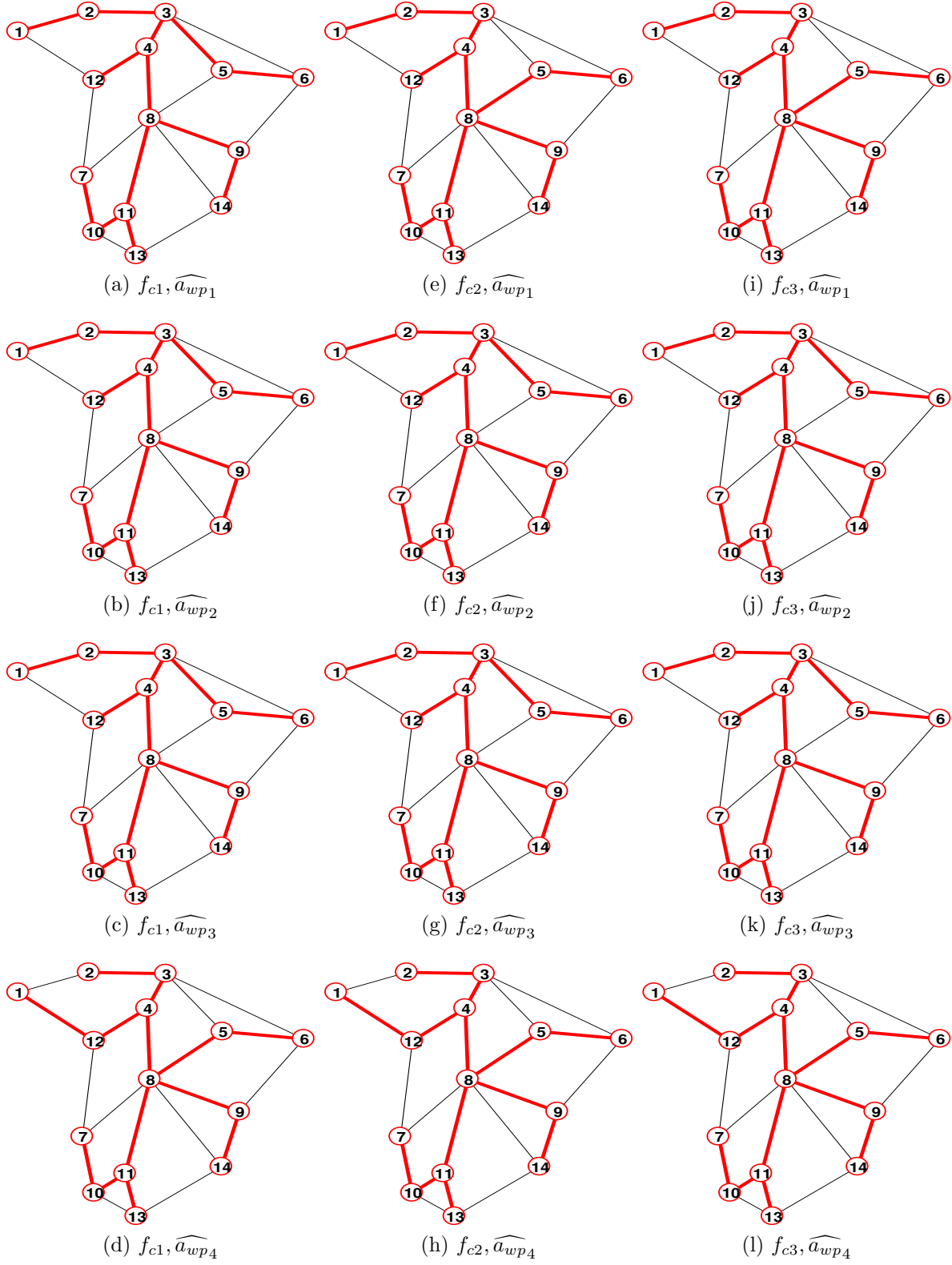


Figure 5.25: Spines obtained for the Spain network using the three cost functions.

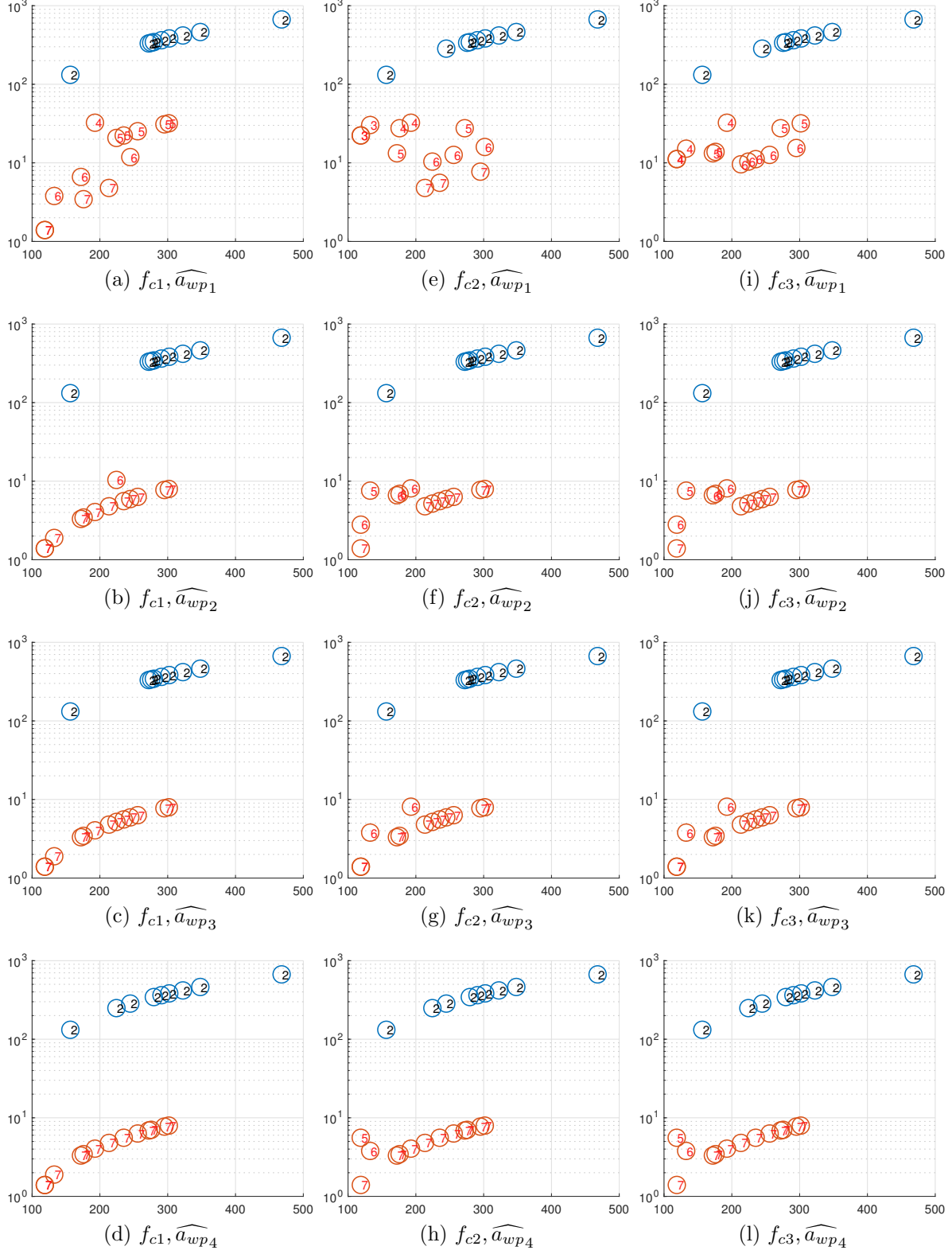
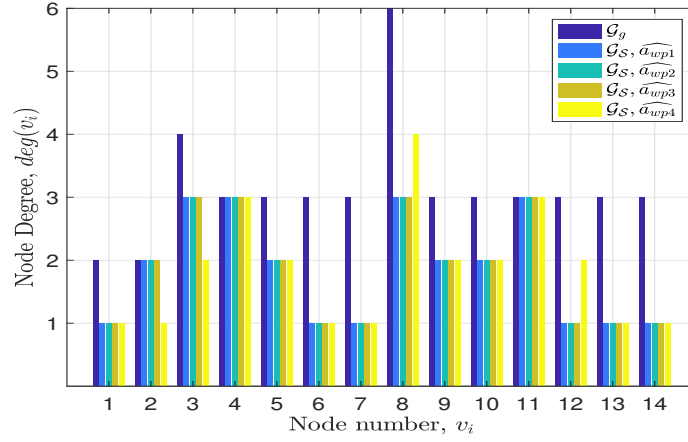
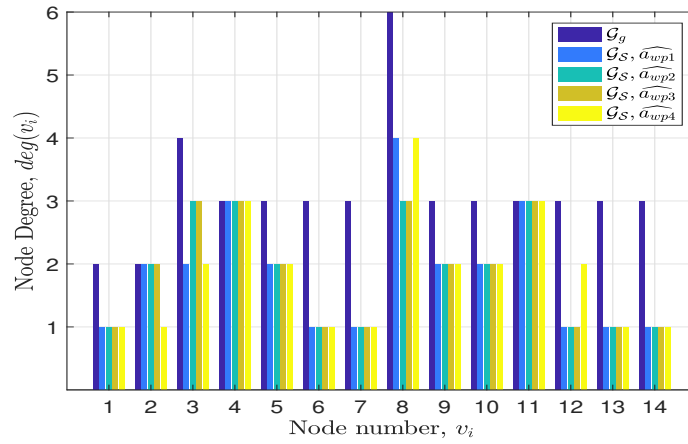


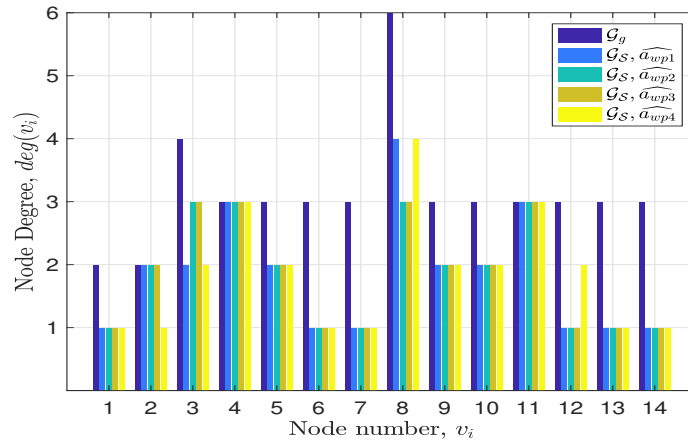
Figure 5.26: The corresponding link downtime/year and versus link length for the spines obtained for the Spain network.



(a) f_{c1} .



(b) f_{c2} .



(c) f_{c3} .

Figure 5.27: Node Degree for the resulted Spain spines.

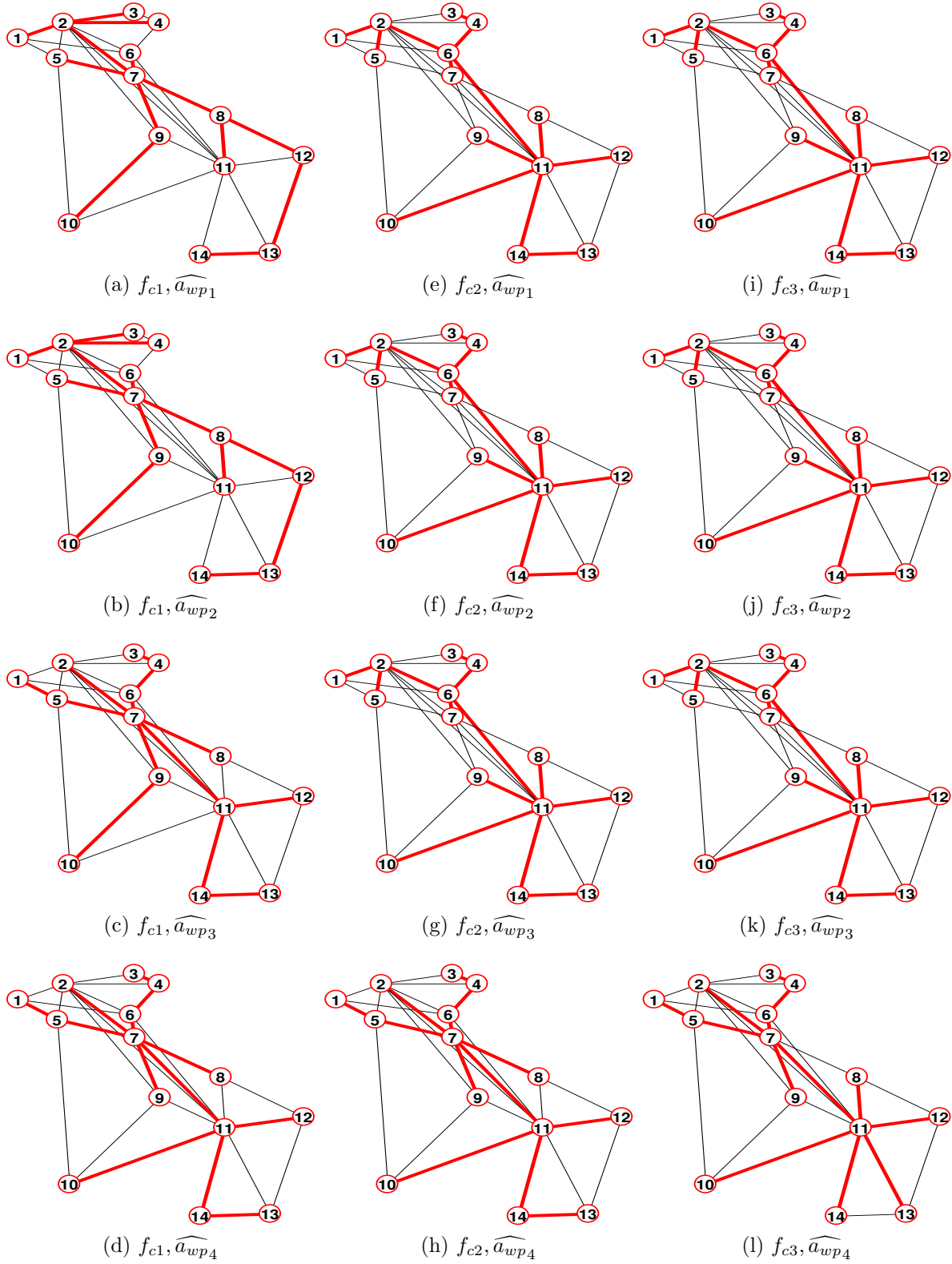


Figure 5.28: Spines obtained for the Italia14 network using the three cost functions.

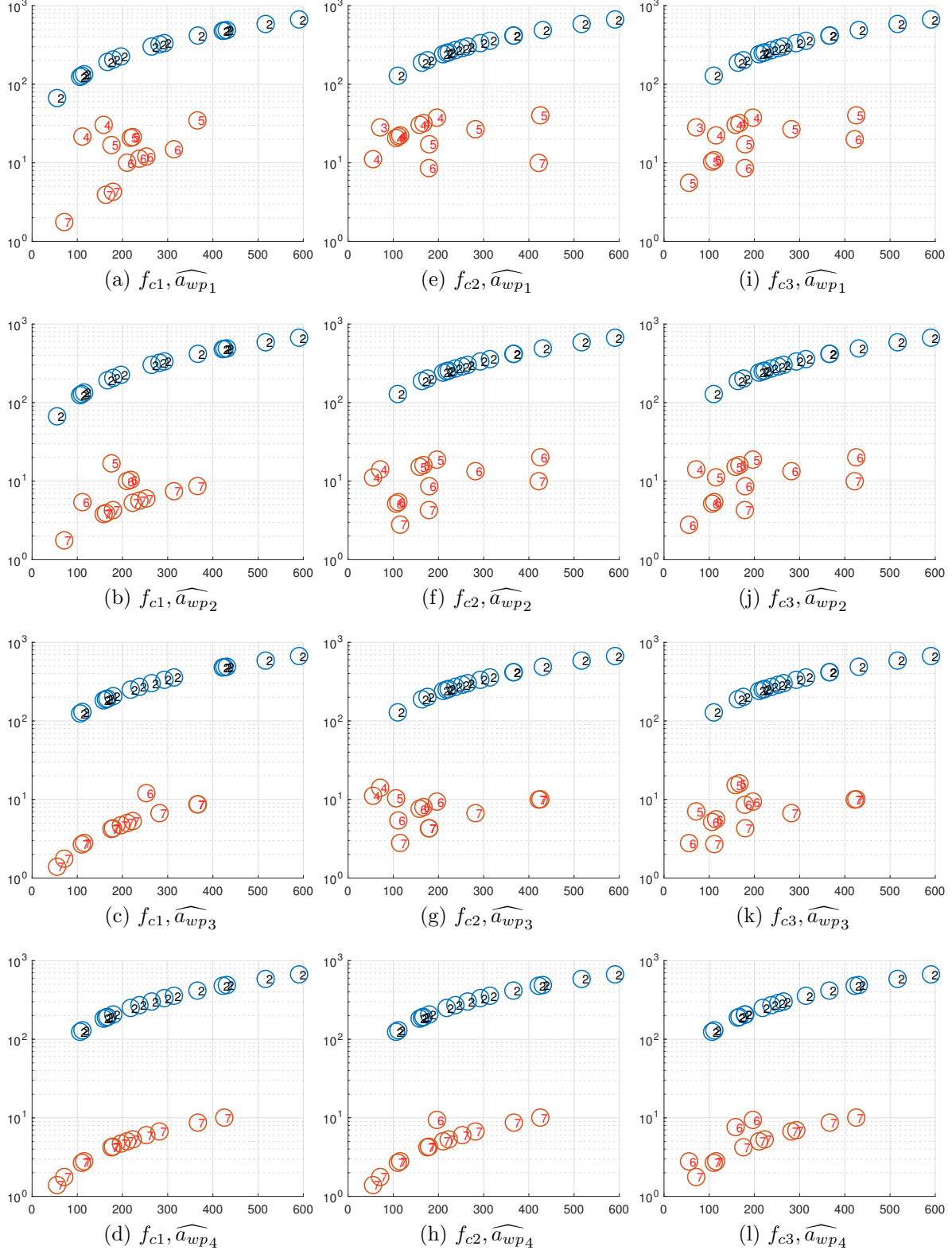
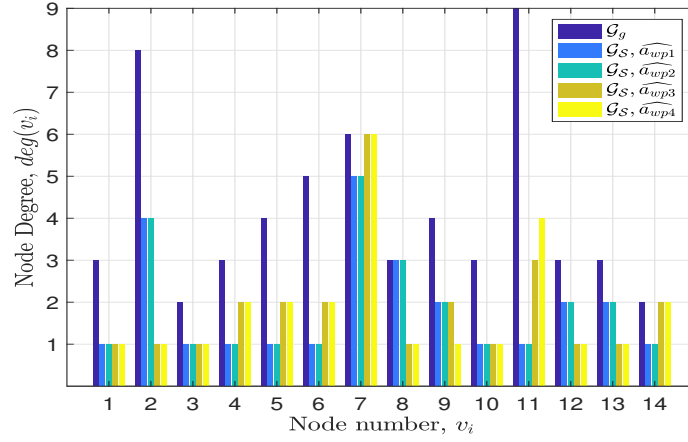
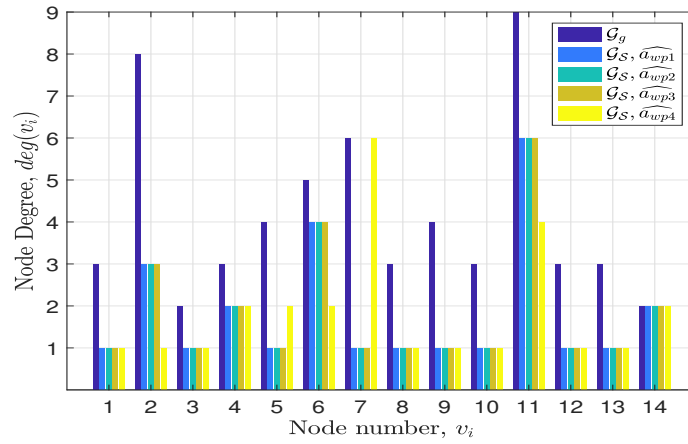


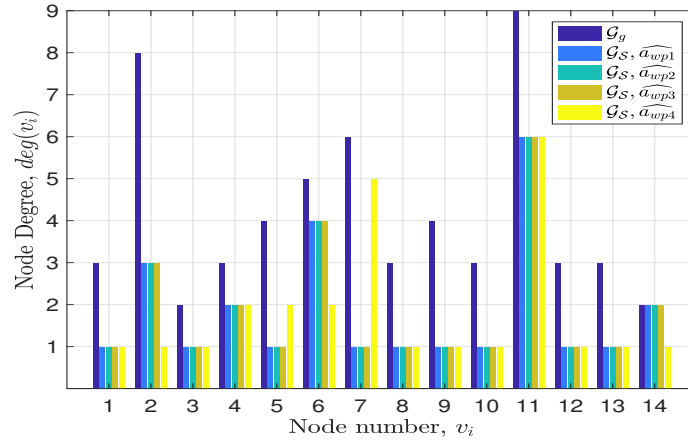
Figure 5.29: The corresponding link downtime/year and versus link length for the spines obtained for the Italia14 network.



(a) f_{c1} .



(b) f_{c2} .



(c) f_{c3} .

Figure 5.30: Node Degree for the resulted italia14 spines.

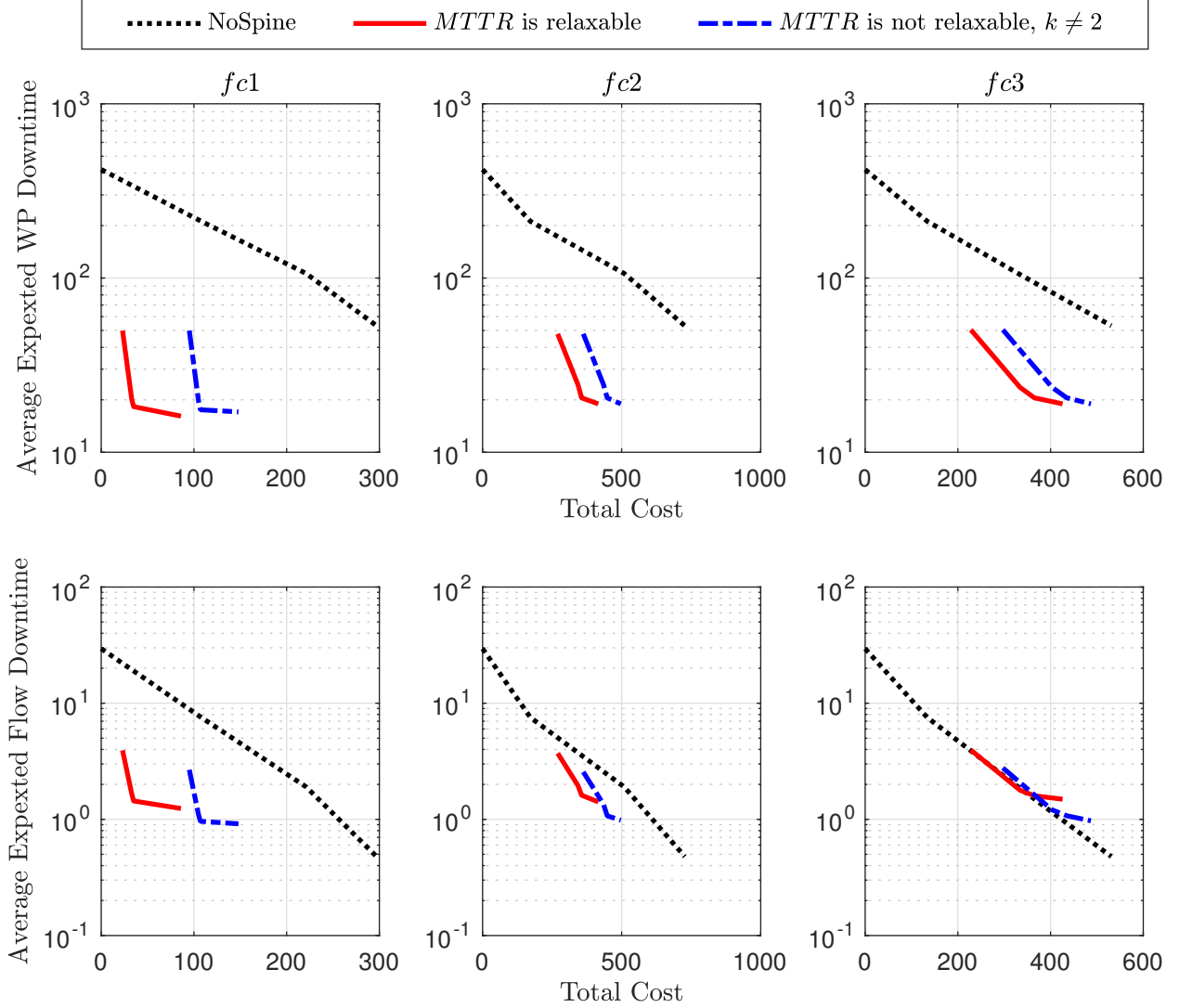


Figure 5.31: Average expected WP and end-to-end flow downtime/year versus cost for the spines obtained for the Polska network.

by a link-disjoint backup path, and its expected downtime is minimal. Within each scenario, the range of availability for the middle class is upper bounded by the target availability, \widehat{a}_{wp} , as shown by the maximum downtime bar. The variation within that range, however, is attributed to the layout of the spine, and in general to the structure of the network, as some connections typically have longer paths on the spine. We expect that a spine with very long diameter to have a wider range of flows downtime. One also can see from the graph that the

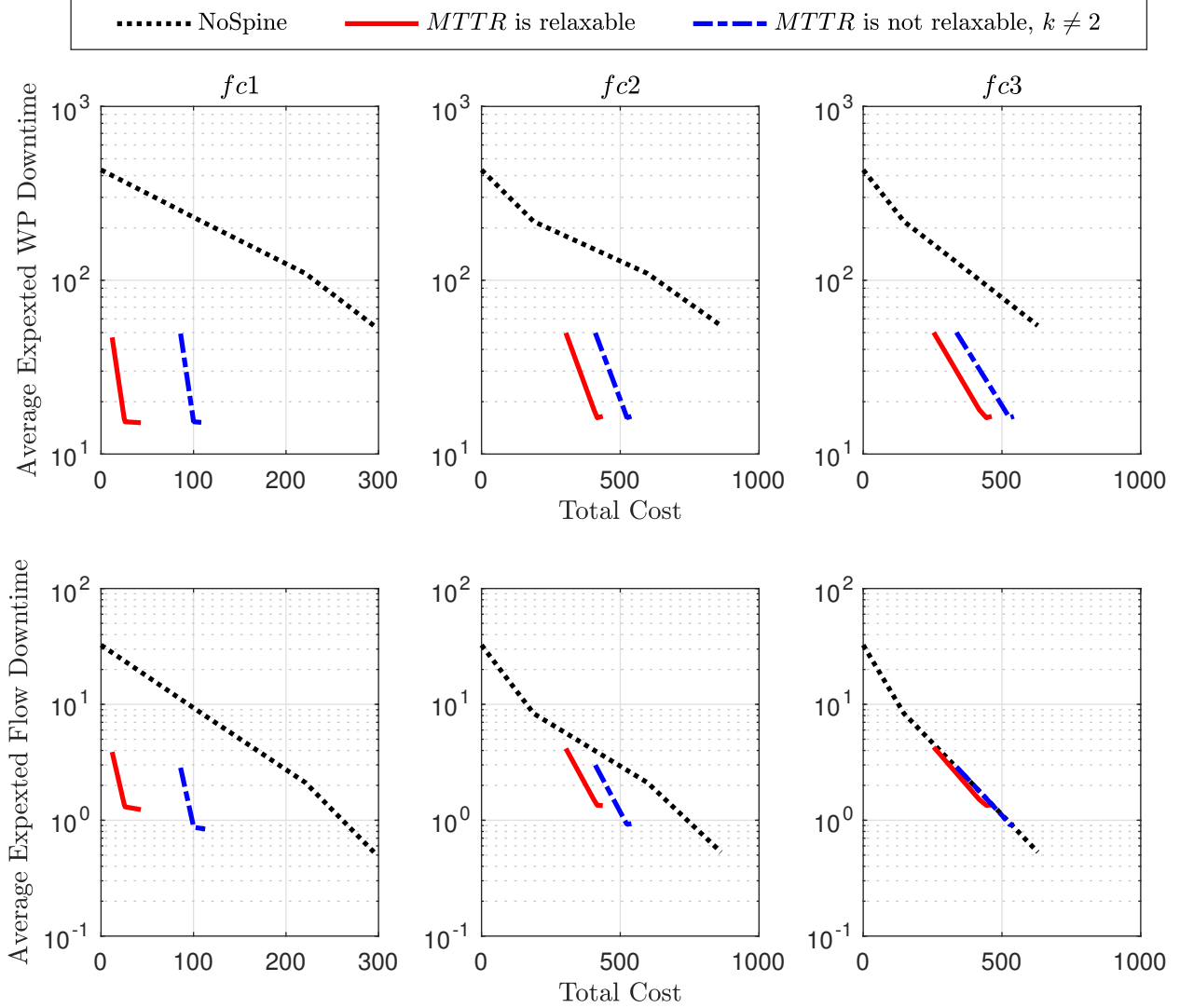


Figure 5.32: Average expected WP and end-to-end flow downtime/year versus cost for the spines obtained for the Spain network.

target availability also controls the downtime of the higher class since the WP of this class is routed on the spine, and its downtime decreases as the target availability increases. The maximum flow downtime within this class is close to the median downtime value compared to the minimum which means that the range of the flows downtime is moderately narrow above the median. This is also can be attributed to the structure of the network but mainly is due to the WP target availability as the end-to-end availability of a path-pair is lower

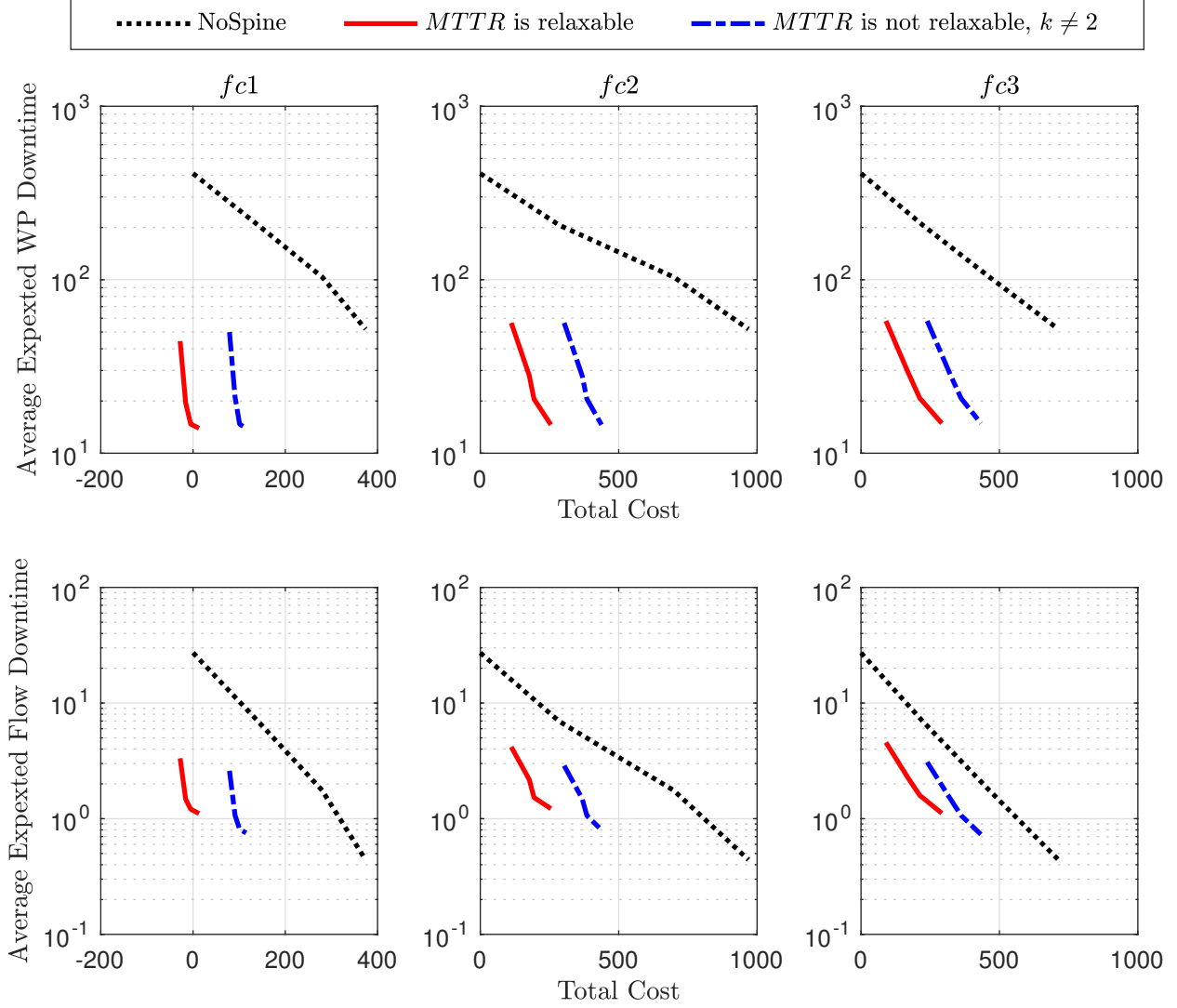


Figure 5.33: Average expected WP and end-to-end flow downtime/year versus cost for the spines obtained for the Italia14 network.

bounded by the availability of the highest path availability. For the lower class, the expected downtime is independent from the WP target availability, and it maintains a similar range of downtimes across the different scenarios. Lastly, the spacing between each level of availability is mainly determined by the range of link availabilities (initial and improved) and the WP target availability.

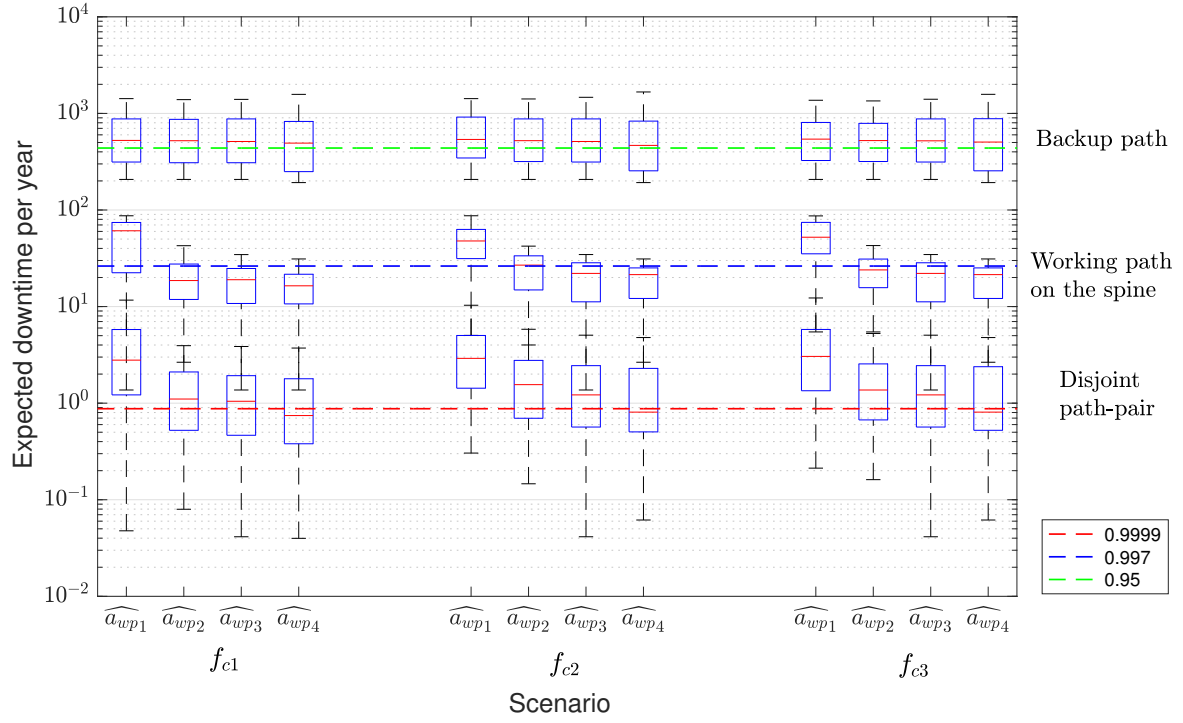


Figure 5.34: The range of paths expected downtime in the spines of the Polska network.

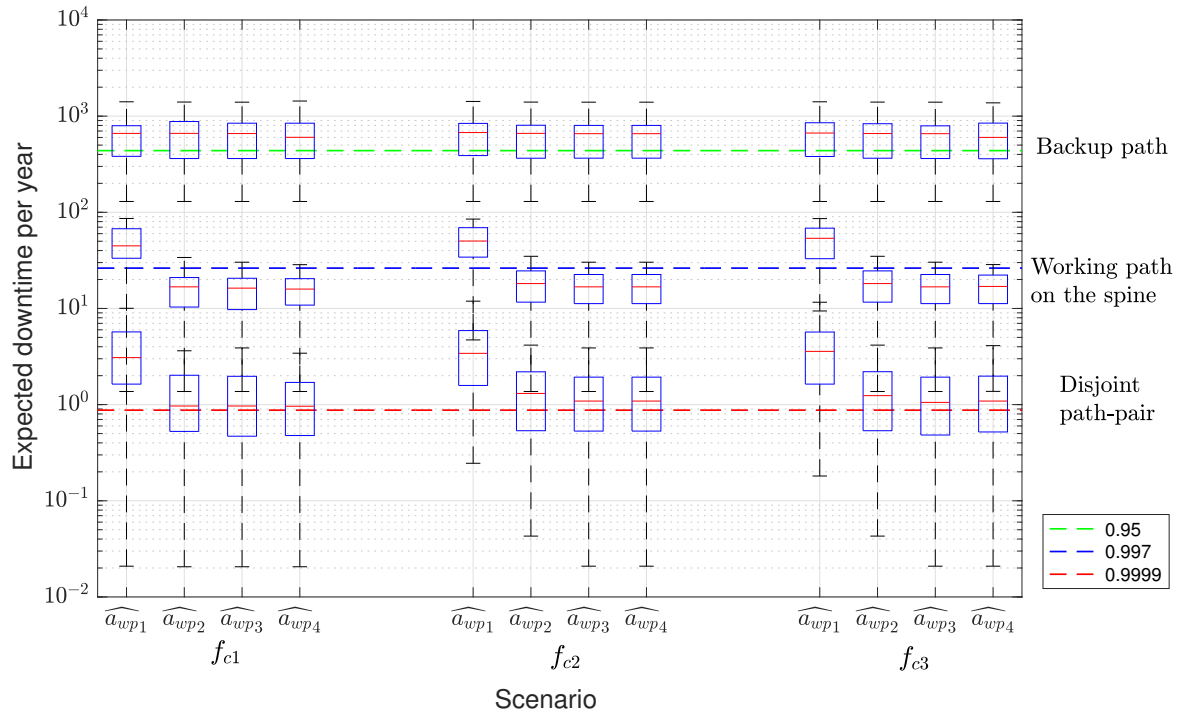


Figure 5.35: The range of paths expected downtime in the spines of the Spain network.

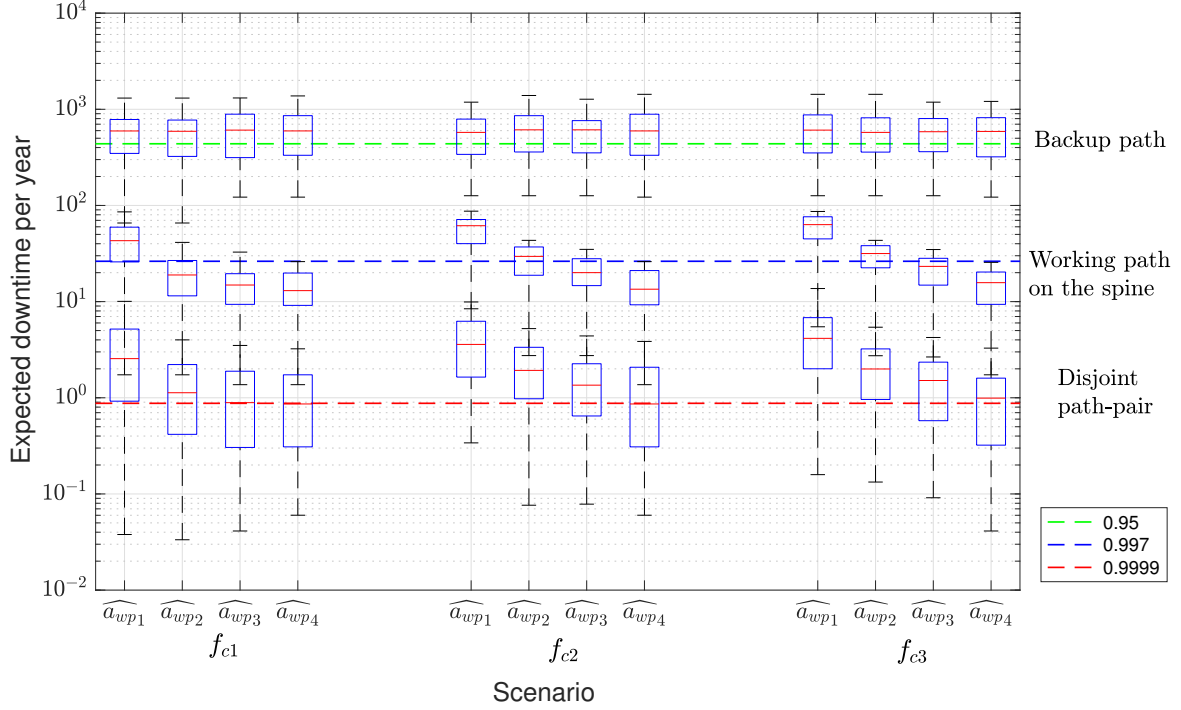


Figure 5.36: The range of paths expected downtime in the spines of the Italia14 network.

5.6 SUMMARY

In this chapter, we presented the spine concept of embedding a subgraph structure with higher availability in a network together with protection mechanisms aiming to improve the overall end-to-end availability. The spine based approach was shown to have the potential to improve the network availability in a more efficient fashion compared to improving the availability of all network components in a homogeneous fashion. We show that the efficiency of the spine is highly influenced by the heterogeneity of link availability, the network size, and the spine layout. We provide heuristic spine selection methods based on structural properties of the network topology. The goal was to find a way to embed a spine to achieve a maximum average end-to-end availability and the results appear promising compared to optimal spine values determined by a brute force search. In addition, we provide an optimization based formulation for designing the spine taking into consideration that links

availability are upgradeable for a given cost. The design problem aims at exploiting existing heterogeneity in link availability and the upgradability of link availability to achieve a target flow availability while minimizing the total cost. Our results demonstrate the spine model efficiency in terms of average flow availability and potential advantage over the shortest path model with no spine. This efficiency, however, depends primarily on network density and link improvement cost distribution. Also the obtained spines have similar structural properties to the spines obtained from the heuristics with slight variations as the cost of improving link availability imposes an additional constraint in shaping the spine layout. In general the spine is hoped to provide larger differences in the range of availability values to quality of resilience classes resulting in less over engineering of the network to meet the most stringent availability requirements.

6.0 THE JOINT CROSSLAYER ROUTING-MAPPING DESIGN PROBLEM

6.1 INTRODUCTION

As discussed earlier in Chapter 3 crosslayer mapping problem has been studied in literature, specifically in the survivable mapping problem context [18, 19]. However, designs considering explicit availability constraints are somewhat uncommon. There have been different strategies to include these constraints. One approach is to minimize the overall crosslayer failure probability by constrained mapping [128], which is equivalent to maximizing the availability of logical links (the sum of the physical links availabilities) [98]. An alternate method is to minimize the failure probability for crosslayer minimum-cuts by augmenting additional logical links [97]. Availability constraints can also be involved in defining other metrics *e.g.*, risk [96], which can be optimized. These proposals, however, are based on network overall availability and do not differentiate between service classes.

In the previous Chapter 5, we implemented the spine concept and proposed to use the spine for routing connections of the higher availability QoR classes. We showed how the spine concept can be utilized to provide levels of availability differentiation at the physical level using constrained and differentiated routing and protection techniques. In addition, the spine can be leveraged with logical network topology routing, cross layer mapping, and protection schemes to differentiate resilience classes and provide a range of availability guarantees. In this chapter, we introduce a joint crosslayer routing-mapping design problem with the aim of providing multiple levels of availability guarantees in multilayer networks using a predefined spine embedded at the physical layer.

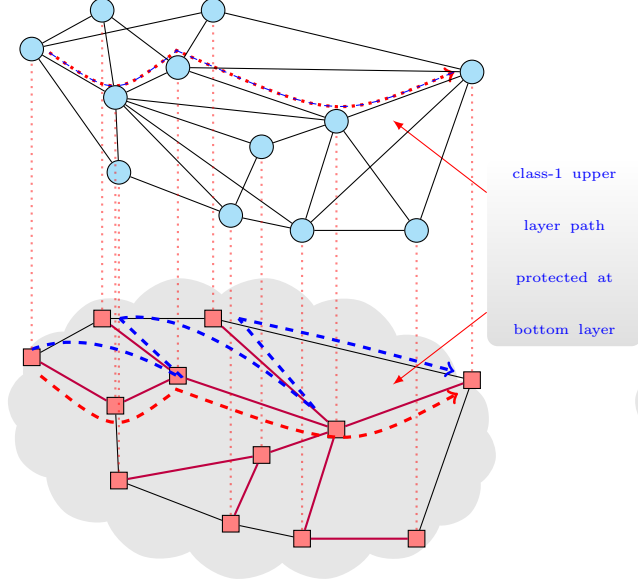


Figure 6.1: Protection at bottom layer.

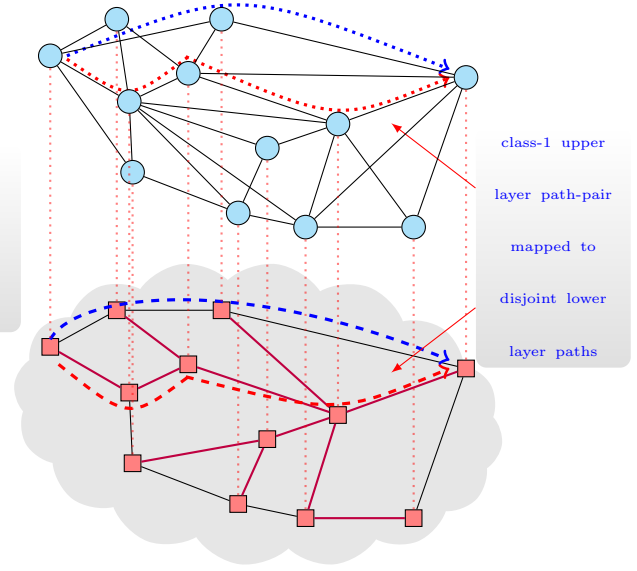


Figure 6.2: Protection at top layer.

6.2 PROTECTION CONFIGURATION

Consider a scenario where different QoS classes have traffic demand between each node pair in the upper layer. Demands should first be routed on the logical layer using the available logical links. Then, each logical link is mapped to a physical path with enough capacity to serve all upper layer demands. The logical routing should isolate the traffic of different classes in order to keep them distinguishable such that the physical network can treat them in a differentiated manner. This would result in multiple logical subnetworks. Some classes can be assigned protection backup paths, and protection can be given at lower or upper layer, as shown in Figures 6.1 and 6.2 respectively.

Let us, for instance, consider two classes of traffic. If protection is given at the lower layer, class-1 traffic can be routed through logical paths that are mapped to the spine and protected by link-disjoint backup paths at the physical layer. The group of logical links dedicated for establishing class-1 paths (*i.e.*, a logical subnetwork) is called the logical spine. Another logical network can be configured to carry another class (call it class-4), and each logical

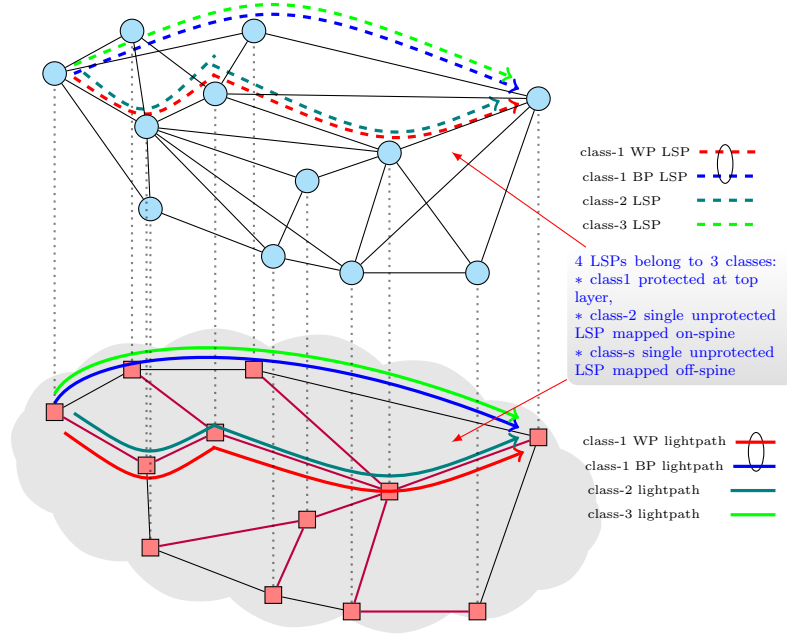


Figure 6.3: Illustration of 3 classes with different configurations with protection, if applicable, given on top layer.

link is mapped, say, into a single path routed freely in the physical layer. Each additional class would require a separate logical network. On the other hand, when protection is given at the upper layer, class-1 traffic working paths can be routed through logical paths that are mapped to the physical layer spine. The group of logical links used for establishing class-1 working paths form the logical spine. When a failure occurs, traffic is rerouted through backup logical paths that are link-disjoint from their associated working paths at both layers. Likewise, the group of logical links used for establishing backup paths is a separate logical subnetwork. That is to say, using two link-disjoint logical subnetworks, one for working paths and the other for backup paths, class-1 traffic can be given 1:1 protection at top layer. Furthermore, each logical subnetwork can be utilized to carry an additional unprotected traffic class. Specifically, one class is carried over the logical spine and another class is carried over the other logical subnetwork. Hence there is a chance to utilize logical links more efficiently. Figure 6.3 shows an illustration of three logical flows routed on two logical subnetworks in a differentiated manner.

Table 6.1: Possible protection configurations for a multilayer network with a spine.

QoR Class	Protection at Bottom Layer	Protection at Top Layer
General configuration	A dedicated logical subnetwork is required for each QoR class.	Only two logical subnetworks are created. One is mapped onto the spine.
1	$P \mapsto (BP : WP^s)$ single logical path mapped to link-disjoint path-pair, either path is routed on spine.	$BP : WP^s \mapsto P : P^s$ logical link-disjoint path-pair routed through links mapped into an underlying disjoint path-pair one of them is on the spine.
2	$P \mapsto BP : WP$ single logical path routed through logical links mapped freely to link-disjoint path-pair.	$BP : WP \mapsto P : P$ logical link-disjoint path-pair routed through links mapped freely into an underlying disjoint path-pair.
3	$P \mapsto P^s$ single logical path routed through links mapped into the spine.	$WP \mapsto P^s$ single logical path routed through the links that are mapped to the spine – the same logical subnetwork used by class-1.
4	$P \mapsto WP$ single logical path routed through links mapped freely.	$WP \mapsto P$ single logical path routed through the links that are mapped freely – the same logical subnetwork used by class-2.

To get more insight into the difference between providing protection at bottom and top layers, Table 6.1 shows the possible configurations for both problems with four classes of traffic. The first class can be given a single logical path mapped to a path-pair with one path on the spine (*i.e.*, protection at lower layer) or can be given two logical paths each mapped to a single underlay path one of them is on the spine (*i.e.*, protection at upper layer). The second class follows the same logic of the first class except it ignores the on-spine restriction. The third class uses a single unprotected on-spine path, whereas the forth class is unprotected but ignores the on-spine restriction. Note that we only consider a dedicated path protection. Further configurations that deploy sharing protection schemes can also be integrated into the design.

6.3 PROTECTION AT LOWER LAYER

In this section, we consider the case in which protection can be given to connections at the lower layer only, as shown in Figure 6.1. We formulate this problem as a mathematical programming problem. Basically, the strategy is to (1) route logical flows on the logical network, (2) aggregate the required bandwidth units on each logical link, and then (3) map each link to an end-to-end physical path in the underlaying layer with enough capacity. The mapping is performed in a differentiated manner based on the protection approach (*e.g.*, number of classes supported, type of protection technique used). This will be explained in the subsequent sections. The network design is evaluated in terms of the upper layer logical links and flows availability and the physical capacity efficiency at the lower layer. First, we summarize our notation as follows.

6.3.1 Notation

Sets:

\mathcal{G}_g	a network graph of type g : $\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$
g	P for physical, L for logical, or S for spine.
$\mathcal{V}_g(\mathcal{E}_g)$	set of network nodes/vertices (links/edges).
\mathcal{D}_ϕ	set of end-to-end demands of class- ϕ .

Indexes:

v	node index.
e	link (edge) index.
d_ϕ^{mn}	an upper layer symmetric demand ($d_\phi^{mn} \in \mathcal{D}_\phi$).
i, j	end nodes of a physical link, $(i, j) \in \mathcal{E}_P$.
s, t	end nodes of a logical link, $(s, t) \in \mathcal{E}_L$.
m, n	end nodes of a flow/demand, $m, n \in \mathcal{V}_L$.
ϕ	the class of traffic.
WP (BP)	Working path (Backup path).
P^s	the superscript s indicates a path/link is routed/mapped on the spine.

\mapsto mapped to.

Capacity:

c_{ij} input parameter indicating the capacity of physical link (i, j) .
 w_{ϕ}^{st} internal variable that measures the capacity on logical link (s, t) due to flows of class ϕ .
 W_{st} input parameter indicating the maximum capacity of logical link (s, t) .

Availability:

a_{ij} availability of link (i, j) .

A_{st}^{WP} Working Path availability for logical link (s, t) :

$$A_{st}^{WP} = \prod_{ij \in WP_{st}} a_{ij} \quad (6.1)$$

A_{st}^{BP} Backup Path availability for logical link (s, t) (similar to equation (6.1)).

A_{st} availability of logical link (s, t) . Assuming WP_{st} and BP_{st} are fully edge-disjoint:

$$A_{st} = 1 - (1 - A_{st}^{WP})(1 - A_{st}^{BP}). \quad (6.2)$$

A_{mn} availability of flow mn .

$$A_{mn} = \prod_{st | mn \mapsto st} A_{st} \quad (6.3)$$

EDT Expected downtime in hours per year. Given availability A :

$$EDT_{hrs/yr} = 24 \times 365 \times (1 - A) \quad (6.4)$$

Variables:

$X_{ij}^{st, \phi}$ a continuous variable denoting the amount of resources provisioned on physical link (i, j) for realizing the WP of the logical link (s, t) carrying demands of class- ϕ .

$Y_{ij}^{st, \phi}$ a continuous variable capturing the amount of resources provisioned on physical link (i, j) for realizing the BP of the logical link (s, t) carrying demands of class- ϕ .

$Z_{st}^{d_{\phi}^{mn}}$ a continuous variable denoting the amount of demand mn of class ϕ routed through logical link (s, t) .

ξ_{st} a binary selector variable which indicates whether a logical link (s, t) is selected ($\xi_{st} = 1$) or not ($\xi_{st} = 0$) (used in the MILP in the second problem).

6.3.2 Network Model

We consider a two layer network composed of the physical network on the bottom represented as \mathcal{G}_P with the set of nodes and links $(\mathcal{V}_P, \mathcal{E}_P)$, and on top the logical network \mathcal{G}_L represented by the set of logical nodes and links $(\mathcal{V}_L, \mathcal{E}_L)$. We assume that each logical node is connected to its corresponding physical node (*i.e.*, $\mathcal{V}_L = \mathcal{V}_P$). Logical links are realized as paths in the physical network. The spine is defined as a subgraph of the physical network, \mathcal{G}_P , with a subset of nodes and links $(\mathcal{V}_S, \mathcal{E}_S)$ in which $\mathcal{V}_S \subseteq \mathcal{V}_P$ and $\mathcal{E}_S \subset \mathcal{E}_P$. To support full connectivity between all node pairs, we select the spine to be a minimum spanning tree which is the least cost connected graph. Hence, $|\mathcal{E}_S|$ is strictly equal to $|\mathcal{V}_P| - 1$. Demands between pairs are given as a demand matrix, D^ϕ , where ϕ is the class identifier. Demands are first routed on the logical layer using the available logical links. Then, each logical link is mapped to a physical path with enough capacity to serve all upper layer demands.

Note that the logical routing should isolate the traffic of different classes in order to keep them distinguishable such that the physical network can treat them in a differentiated manner. This would result in multiple logical networks, one for each class. In this part, we consider two classes: flows of class-1 ($\phi = 1$) which require high availability levels while class-4 ($\phi = 4$) are of a lesser importance and have no strict availability requirements. Therefore, the flows of class-1 are routed on logical links that are mapped to a fully disjoint working and backup path-pair at the physical network, one of which is restricted to be on the spine, as shown in Figure 6.1. In contrast, the flows of class-4 are routed freely on the network with no protection. In the following section, we present two different formulations of the design problem.

6.3.3 Formulation

In this section, we formulate the QoR multilayer design problem for two approaches to isolating the flows of the different classes.

6.3.3.1 Problem I: LP for duplicated logical links In the first problem, we assume that each logical link is duplicated and each class would have a set of logical links that it

can utilize exclusively. We formulate this problem as a Linear Programming (LP) problem given by (6.5)-(6.14).

$$\begin{aligned} & \underset{X_{ij}^{st,\phi}, Y_{ij}^{st,\phi}}{\text{minimize}} && \sum_{\phi} \sum_{ij \in \mathcal{E}_P} \sum_{st \in \mathcal{E}_L} (X_{ij}^{st,\phi} + Y_{ij}^{st,\phi}) \\ & \text{s.t.} && \end{aligned} \quad (6.5)$$

$$\sum_{kt \in \mathcal{E}_L} Z_{kt}^{d_\phi^{mn}} - \sum_{sk \in \mathcal{E}_L} Z_{sk}^{d_\phi^{mn}} = \begin{cases} d_\phi^{mn} & \text{if } k = m \\ -d_\phi^{mn} & \text{if } k = n \\ 0 & \text{otherwise} \end{cases}, \forall k \in \mathcal{V}_L, d_\phi^{mn} \quad (6.6)$$

$$\sum_{ik \in \mathcal{E}_S} X_{ik}^{st,\phi} - \sum_{kj \in \mathcal{E}_S} X_{kj}^{st,\phi} = \begin{cases} w_\phi^{st} & \text{if } k = s \\ -w_\phi^{st} & \text{if } k = t \\ 0 & \text{otherwise} \end{cases}, \forall k \in \mathcal{V}_S, \phi = 1 \quad (6.7)$$

$$\sum_{ik \in \mathcal{E}_P} X_{ik}^{st,\phi} - \sum_{kj \in \mathcal{E}_P} X_{kj}^{st,\phi} = \begin{cases} w_\phi^{st} & \text{if } k = s \\ -w_\phi^{st} & \text{if } k = t \\ 0 & \text{otherwise} \end{cases}, \forall k \in \mathcal{V}_P, \phi \neq 1 \quad (6.8)$$

$$\sum_{ik \in \mathcal{E}_P} Y_{ik}^{st,\phi} - \sum_{kj \in \mathcal{E}_P} Y_{kj}^{st,\phi} = \begin{cases} w_\phi^{st} & \text{if } k = s \\ -w_\phi^{st} & \text{if } k = t \\ 0 & \text{otherwise} \end{cases}, \forall k \in \mathcal{V}_P, \phi = 1 \quad (6.9)$$

$$X_{ij}^{st,\phi} + Y_{ij}^{st,\phi} \leq w_\phi^{st}, \quad \forall (i, j) \in \mathcal{E}_P, (s, t) \in \mathcal{E}_L, \phi \quad (6.10)$$

$$\sum_{st \in \mathcal{E}_L} \sum_{\phi} (X_{ij}^{st,\phi} + Y_{ij}^{st,\phi}) \leq c_{ij}, \quad \forall (i, j) \in \mathcal{E}_P \quad (6.11)$$

$$\sum_{mn} \sum_{\phi} Z_{st}^{d_\phi^{mn}} \leq W_{st}, \quad \forall (s, t) \in \mathcal{E}_L \quad (6.12)$$

$$\sum_{mn} Z_{st}^{d_{\phi}^{mn}} = w_{\phi}^{st}, \quad \forall (s, t) \in \mathcal{E}_L, \phi \quad (6.13)$$

$$X_{ij}^{st,\phi} \geq 0, \quad Y_{ij}^{st,\phi} \geq 0, \quad Z_{st}^{d_{\phi}^{mn}} \geq 0 \quad (6.14)$$

The objective of the model (6.5) is to minimize the total physical resources used by all logical links of all classes when carrying all flows. The first sets of constraints (6.6)-(6.9) are the classical network flow conservation in the node-link formulation, namely: constraint (6.6) deals with routing the demands at the logical layer, constraint (6.7) maps the total capacity required by a logical link carrying class-1 demands onto a physical path on the spine, constraint (6.8) deals with mapping class-4 logical links with no restriction, and lastly constraint (6.9) maps a class-1 logical link into a redundant backup path. Note that in the duplicate problem, variables $Z_{st}^{d_{\phi=1}^{mn}}$ and $Z_{st}^{d_{\phi=4}^{mn}}$ used for routing demands of class-1 and class-4 are independent and each can utilize a similar set of logical links. Constraint (6.10) ensures that a working-backup path-pair realizing a logical link is disjoint. Specifically, for each logical link (s, t) , it only allows one variable, either a working path (WP) or a backup path (BP) variable, to provision the capacity of the logical link, w_{ϕ}^{st} . Constraints (6.11) and (6.12) ensure that the aggregate capacity provisioned on physical and logical links, respectively, are below their maximum limits. Constraint (6.13) determines the aggregate capacity of class ϕ demand that is routed on a logical link. Lastly, constraint set (6.14) defines the non-negativity requirements of the decision variables as described in the notation Section 6.3.1.

6.3.3.2 Problem Type II: MILP for partitioned logical network In contrast to the first method, in this case, the logical networks of the different classes do not necessarily need to be identical. Instead, the logical network is partitioned into two subnetworks, each is capable of carrying all demands of a particular class. In the first method, where each logical link in the network is duplicated for each class may not be efficient since the logical links of each class will have different mapping considerations (*e.g.*, on spine, protected, etc.) and consequently they might have different routing preferences in the logical layer. Hence splitting up the set of logical links into subsets based on class can be a more reasonable way to achieve separation between different classes.

By slightly modifying the previous LP, we can formulate this approach as a Mixed Integer Linear Programming (MILP) problem. Specifically, we define a binary selector variable ξ_{st} , and add the following constraints:

$$\sum_{mn} Z_{st}^{d_{\phi}^{mn}} - M\xi_{st} \leq 0, \forall (s, t) \in \mathcal{E}_L, \phi = 1 \quad (6.15)$$

$$\sum_{mn} Z_{st}^{d_{\phi}^{mn}} - M(1 - \xi_{st}) \leq 0, \forall (s, t) \in \mathcal{E}_L, \phi = 4 \quad (6.16)$$

$$\xi_{st} \in (0, 1) \text{ is binary} \quad (s, t) \in \mathcal{E}_L \quad (6.17)$$

where M is a large constant value. This ensures that when ξ_{st} is equal to 1 the logical link can only be used by class-1 demands, otherwise it can only be used by class-4 demands.

6.3.4 Numerical Study

6.3.4.1 Scenarios We consider the Polska, NSF, Spain, EPAN, and Italia14 networks for our numerical study. Table 6.2 shows the number of nodes and links in the physical layer of these networks. For each physical network, we use three spines from the spines computed

Table 6.2: Test Networks.

G_P	Network	$ \mathcal{V}_P $	$ \mathcal{E}_P $	$\frac{ \mathcal{E}_P }{ \mathcal{V}_P }$	diameter	Spine diameter		
						S_1	S_2	S_3
1	Polska [Table 5.3]	12	18	1.50	4	8	5	6
2	NSF [Table 5.5]	14	19	1.36	4	7	6	9
3	Spain [Table 5.5]	14	22	1.57	5	6	5	7
4	Italia14 [Table 5.3]	14	29	2.07	3	4	3	4
5	EPAN [Table 5.5]	16	23	1.44	6	8	7	11

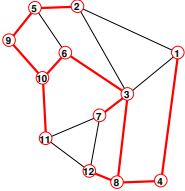
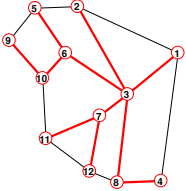
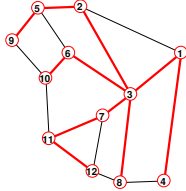
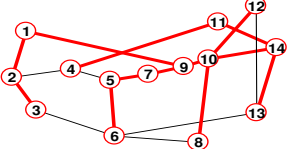
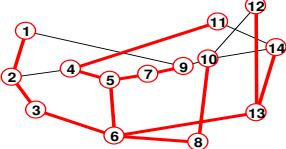
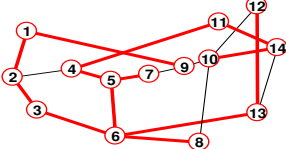
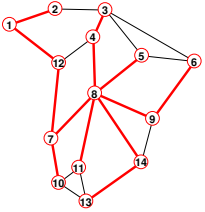
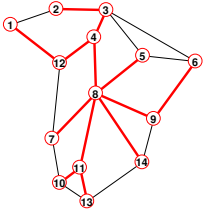
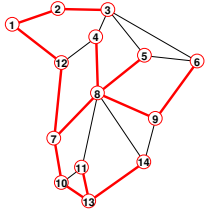
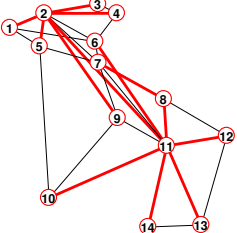
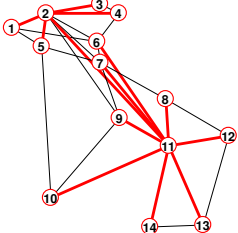
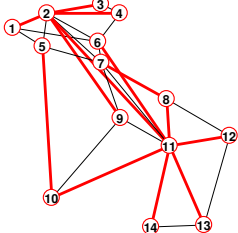
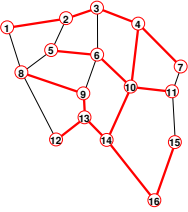
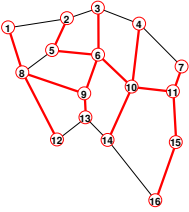
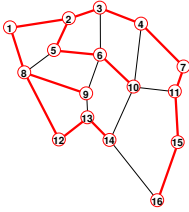
G_P	Network	S_1	S_2	S_3
1	Polska			
2	NSF			
3	Spain			
4	Italia14			
5	EPAN			

Figure 6.4: The layout of the spines used for the test networks.

in Table 5.3 for the Polska and Italia14 networks and in Table 5.5 for the NSF, Spain, and EPAN networks. The layouts of these spines are shown in Figure 6.4. The first spine, \mathcal{G}_{S_1} , is the spine that maximizes the average WP-BP path-pair availability A_S , with the WP routed on the spine and the BP is routed freely on the network and conditioned to be disjoint from the WP. The second spine, \mathcal{G}_{S_2} , maximizes the average WP path availability on the spine A_S^{WP} . Since the spine with largest A_S^{WP} rarely corresponds to the spine with largest A_S , the third spine, \mathcal{G}_{S_3} , was picked as a compromise solution having availability in between the spines \mathcal{G}_{S_1} and \mathcal{G}_{S_2} as discussed in Section 5.3.3. For example, the physical Polska network, which consists of 12 nodes and 18 links, is shown in the bottom layer of Figure 6.5 with red links (in bold) representing a spine, black links represent links that are not part of the spine, and the top part of each figure is a logical network. Note that the spines in Figures 6.5a, 6.5b, and 6.5c have a diameter of 8, 5 and 6, respectively. Hence, although the third spine will result in a slight reduction on the average end-to-end availability A_S with respect to the first spine, its paths will require less resources than the paths in the first spine and its A_S^{WP} value is close to that of the second spine (see Table 5.3). In general the compromise spine can be determined in a number of ways such as optimizing a weighted combination of A_S and A_S^{WP} or maximizing the minimum flow availability (this is the case for NSF and EPAN, see Table 5.5).

For the logical layer, we generate a number of k -regular random graphs with the same set of physical nodes, $\mathcal{V}_L = \mathcal{V}_P$, using $k = 3, 4, 5, 6$, and 7. These random graphs can be completely random, as in Figure 6.5a, or random with a preselected set of links, as shown in Figure 6.5b where links in bold blue are preselected to match the links on the spine.

For each spine, we ran 14 scenarios that are shown on the left side of Table 6.3 where S_1, S_2 and S_3 refer to the spines shown in Figure 6.4 for each network. These scenarios vary in a number of criteria: node degree of the k -regular graph, whether the graph is completely random or not, and the problem type, duplicate or partitioned logical network. Each scenario is repeated 7 times and the results are averaged. Further, we consider a full mesh of upper layer flows with a single unit bandwidth demand for each class ($d_\phi^{mn} = 1, \forall \phi, mn$).

We compare the averaged results in terms of resource use, logical link availability per class, and end-to-end flow availability per class. Resource use is the amount of reserved

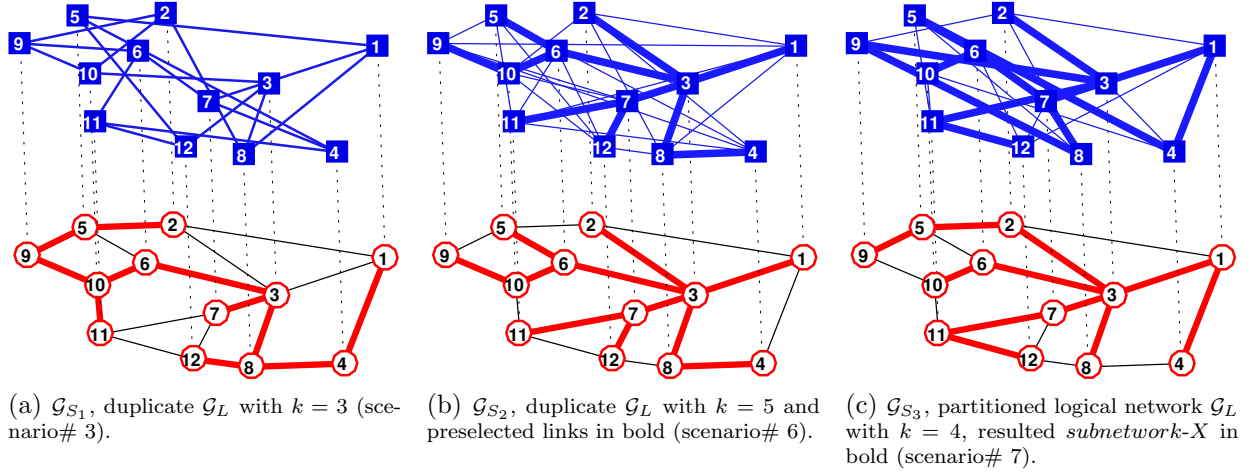


Figure 6.5: Polska network with different spines and logical networks.

physical capacity required to realize the logical links required bandwidth. This is given by the objective function (6.5) in the optimization problem. We assume unlimited bandwidth capacity at both the physical and logical layers, effectively relaxing constraints (6.11) and (6.12). Then mapping and routing information are used to compute logical link availability A_{st} and logical flow availability A_{mn} for each link and flow using equations (6.1)-(6.3). After computing individual link and flow availabilities, we use the average values $\bar{A}_{st} = \sum_{st \in \mathcal{E}_L} A_{st} / |\mathcal{E}_L|$ and $\bar{A}_{mn} = \sum_{d_{\phi}^{mn} \in \mathcal{D}_{\phi}} A_{mn} / |\mathcal{V}_L| \times (|\mathcal{V}_L| - 1) / 2$ as metrics for comparison. Additionally we compare the minimum link $A_{st}^{\min} = \min_{st \in \mathcal{E}_L} A_{st}$ and minimum flow availabilities $A_{mn}^{\min} = \min_{d_{\phi}^{mn} \in \mathcal{D}_{\phi}} A_{mn}$ which determine the maximum expected downtime in hours per year as in equation (6.4).

6.3.4.2 Results We use AMPL/CPLEX and AMPL/Gurobi on NEOS server [129–131] to solve our models. The results are with 0% integrality gap for the duplicate scenarios and within a 1% integrality gap for the partitioned logical network scenarios. Instances that require computing resources more than the NEOS server offers are solved using Gurobi solver on a local computer. The main objective of our design is to minimize the total physical resources used for realizing logical links carrying demands of both classes. Table 6.3 shows

Table 6.3: Average resource use for the three spines in all scenarios.

Network	Scenario	Problem type	regular graph degree k	total no. of logical links	preselected logical links	Average total resources			Average resources for class-1 WP			Average resources for class-4 WP			Average resources for class-1 BP		
						S_1	S_2	S_3	S_1	S_2	S_3	S_1	S_2	S_3	S_1	S_2	S_3
Polska	1	duplicated logical network	3	36	no	1119	1008	1047	433	342	375	275	275	275	411	391	397
	2		3	36	yes	860	748	761	268	202	201	174	168	174	417	379	385
	3		4	48	no	890	813	835	329	280	296	218	218	218	343	315	320
	4		4	48	yes	765	679	697	258	205	207	164	160	164	343	313	326
	5		5	60	no	748	675	698	282	231	246	180	180	180	285	264	272
	6		5	60	yes	700	635	642	250	205	207	152	154	154	297	276	281
	7	partitioned logical network	4	24	no	1332	1232	1268	452	393	408	386	400	391	494	438	469
	8		4	24	yes	1152	1063	1081	316	318	283	342	316	336	495	430	462
	9		5	30	no	1002	929	951	333	293	304	298	292	287	371	344	360
	10		5	30	yes	939	880	884	302	295	261	255	253	271	383	332	352
	11		6	36	no	866	811	838	298	269	280	243	248	250	325	295	308
	12		6	36	yes	830	747	804	284	264	245	220	199	233	326	284	327
	13		7	42	no	803	740	762	294	247	260	223	221	226	286	273	276
	14		7	42	yes	782	710	723	272	242	230	210	196	206	300	272	288
NSF	1	duplicated logical network	3	42	no	1743	1716	1780	650	635	665	419	419	430	674	662	685
	2		3	42	yes	1429	1360	1456	408	384	436	258	250	258	764	726	763
	3		4	56	no	1412	1366	1438	514	484	513	340	340	340	558	543	585
	4		4	56	yes	1235	1197	1266	393	385	407	242	235	237	600	577	621
	5		5	70	no	1237	1196	1263	444	429	447	291	291	291	502	476	524
	6		5	70	yes	1145	1108	1167	420	361	393	232	232	237	493	515	537
	7	partitioned logical network	4	28	no	2146	2108	2204	782	770	727	533	544	604	831	794	873
	8		4	28	yes	1915	1867	1970	601	625	570	442	409	450	872	833	950
	9		5	35	no	1665	1612	1695	590	598	575	435	423	455	640	592	666
	10		5	35	yes	1503	1470	1548	517	516	501	359	364	384	627	589	663
	11		6	42	no	1413	1381	1460	500	487	490	377	379	391	537	515	579
	12		6	42	yes	1350	1316	1381	492	458	467	308	319	335	550	539	579
	13		7	49	no	1322	1287	1359	463	452	494	346	350	351	513	485	514
	14		7	49	yes	1245	1205	1275	452	430	459	299	288	304	494	487	513
Spain	1	duplicated logical network	3	42	no	1658	1613	1765	586	544	687	439	439	439	633	630	638
	2		3	42	yes	1128	1129	1230	315	290	383	238	242	255	575	597	593
	3		4	56	no	1238	1206	1290	437	415	512	318	318	318	482	473	459
	4		4	56	yes	1049	1043	1121	307	294	370	233	237	245	509	512	507
	5		5	70	no	1132	1090	1172	399	363	440	293	293	293	440	434	439
	6		5	70	yes	998	970	1045	303	307	363	230	226	234	466	437	447
	7	partitioned logical network	4	28	no	1940	1875	1997	613	601	668	597	587	604	730	686	725
	8		4	28	yes	1629	1599	1725	401	376	507	504	523	474	724	700	744
	9		5	35	no	1535	1483	1577	499	448	527	468	497	497	567	538	553
	10		5	35	yes	1383	1321	1422	361	423	469	431	353	391	590	545	561
	11		6	42	no	1336	1299	1374	414	410	449	417	412	431	505	477	494
	12		6	42	yes	1238	1194	1272	394	391	435	335	336	354	509	466	483
	13		7	49	no	1172	1154	1218	395	384	417	343	342	352	433	429	449
	14		7	49	yes	1115	1087	1186	367	361	384	301	291	334	446	435	468
Italia14	1	duplicated logical network	3	42	no	1367	1408	1385	473	463	486	379	391	379	514	554	520
	2		3	42	yes	828	778	869	222	202	231	208	196	215	398	380	423
	3		4	56	no	1063	1011	1050	374	353	385	286	277	277	404	381	388
	4		4	56	yes	796	767	826	225	203	242	201	196	206	370	368	378
	5		5	70	no	914	886	925	324	302	333	244	244	244	347	340	348
	6		5	70	yes	775	740	793	230	211	241	199	190	201	347	339	351
	7	partitioned logical network	4	28	no	1640	1598	1643	522	508	519	538	546	550	580	544	574
	8		4	28	yes	1348	1235	1257	365	262	307	458	527	440	525	446	510
	9		5	35	no	1264	1236	1270	388	371	402	411	410	417	465	455	451
	10		5	35	yes	1083	1020	1110	269	247	275	377	349	375	437	424	460
	11		6	42	no	1082	1067	1093	337	334	337	350	348	353	394	385	404
	12		6	42	yes	984	928	1009	257	228	267	318	307	334	409	393	408
	13		7	49	no	947	930	956	313	296	315	291	294	293	343	340	348
	14		7	49	yes	916	875	940	245	218	260	296	284	304	375	374	376
EPAN	1	duplicated logical network	3	48	no	2720	2643	2866	962	903	1122	726	726	726	1032	1014	1018
	2		3	48	yes	1890	1807	1952	513	489	660	382	376	386	995	942	906
	3		4	64	no	2083	2022	2240	719	681	913	541	542	541	823	800	786
	4		4	64	yes	1674	1630	1752	518	488	601	351	358	381	805	784	771
	5		5	80	no	1884	1792	1989	649	620	797	490	490	490	745	682	702
	6		5	80	yes	1572	1521	1645	495	466	594	354	352	361	723	703	690
	7	partitioned logical network	4	32	no	3272	3195	3419	1010	1010	1169	1064	1032	1023	1198	1153	1227
	8		4	32	yes	2746	2669	2676	816	721	723	701	804	772	1228	1144	1180
	9		5	40	no	2561	2452	2643	783	763	953	804	817	806	974	872	884
	10		5	40	yes	2256	2141	2197	677	638	673	652	615	643	927	888	881
	11		6	48	no	2122	2056	2215	712	656	778	635	630	653	774	769	784
	12		6	48	yes	1929	1888	1964	638	606	656	517	485	542	774	797	766
	13		7	56	no	1928	1859	1985	595	611	675	599	581	586	734	667	724
	14		7	56	yes	1818	1715	1861	576	582	625	484	446	517	758	687	719

the resources used by each class for all scenarios. Examining the results, we can make a number of observations as follows.

First, the resource use decreases as the logical network gets denser. For example, in all networks, scenario 6 with the largest number of logical links yields the least resource use. This is because by adding more logical links, we actually increase the probability of finding a shorter logical route/physical mapping combination. Second, there is a slight decrease in resources when spine links are preselected in the logical network. This is more apparent in the results for class-1 and class-4 WPs. For example, there is a significant difference between resource use in scenario 1 and 2 in all networks and for all the three spines. The difference diminishes as the logical network gets denser. The spine, typically, does not follow the shortest path as we cannot combine all shortest paths in a single spanning tree. Since class-1 logical links are restricted to be mapped to paths on the spine, it is likely that the spine would incur more resources, however this depends on the spine's design. In order to evaluate the spines under study in terms of resource efficiency, we compare the resources used for class-1 WPs to resources used by class-4 logical links. We expect the latter to be mapped to the shortest physical path, but the end results of resource usage is also affected by the layout of the logical layer. Class-1 WPs resources are rarely less than class-4 resources and only occurs in the partitioned logical network scenarios. For example, for the Polska network the percentages of the difference between the resources used by class-1 WPs on the spine and class-4 range from 64% to -8% for S_1 , 33% to -2% for S_2 , and 37% to -16% for S_3 . A third observation can be made comparing the resource utilization between the two problem types. In order to compare the two types properly, we consider similar scenarios with comparable number of logical links: specifically scenarios 1 and 2 with 36 logical links to scenarios 11 and 12, respectively. Similarly, 3 and 4 to 13 and 14. Take the Polska network as an example, when the logical network is completely random (*i.e.*, scenario 1 versus 11 & 3 versus 13), the partitioned network problem scenarios require, on average, 9.2-20.7% less resources than the problem with duplicate links. In contrast, the difference becomes negligible when some links are preselected (*i.e.*, scenario 2 versus 12 & 4 versus 14). Though, the latter case might vary across networks *e.g.*, for Spain and Italia14, scenario 12 requires more resources than scenario 2.

After solving the optimization problems, we use path and mapping information to calculate the logical links and flows expected downtimes. We only consider links availability (*i.e.*, assuming nodes are perfect). As noted earlier we use $a_e=0.999$ for links on the spine ($e \in \mathcal{E}_S$) and $a_e=0.99$ for links off the spine ($e \in \mathcal{E}_P$ but $e \notin \mathcal{E}_S$). Table 6.4 shows the average and maximum expected logical link downtime for each class and for each scenario and spine. Similarly, Table 6.5 shows the average and maximum expected flow downtime for each class. First, the same observation made about logical graph density and resource efficiency holds true with respect to flow downtime values. Second, it is clear from both tables that class-1 links and flows achieve much smaller downtime values with averages less than an hour per year compared to class-4 averages which are in orders of tens of hours per year. Also the spine can improve network and services availability differentiation by widening the spread of availability values between the two classes. Note, that this was achieved using only a single backup path protection with no need to using higher configurations (*e.g.*, 1+2, or 1+3). Hence the approach easily enables QoR class differentiation.

To get more insight into the resource use and flow availability of the spine model, we compare our results to a baseline model. The model assumes homogenous link availability at the physical layer in which no spine is embedded and all links have the same availability a_e . In order to have an equivalent network with a comparable cost, a_e is set to a weighted average of the on-spine and off-spine availability, $a_e = \frac{0.999|\mathcal{E}_S|+0.99|\mathcal{E}-\mathcal{E}_S|}{|\mathcal{E}|}$. Logical links are not constrained to be mapped to a spine, and this results in replacing \mathcal{E}_s in our LP formulation by \mathcal{E}_p , with everything else remaining the same. Class-1 logical links are mapped to WP-BP path-pairs, *i.e.*, 1+1 protection, class-4 links are mapped to single paths, and flows are routed on logical links based on their classes. We solve this model for six of the aforementioned scenarios, and compare them to the results of the spine model in Tables 6.3 and 6.5. First, Table 6.6 shows the total resource use for the no-spine model and the percentage of increase in these resources required by the spine model. The percentage of resource increase across all networks ranges from 0.4% to 13.2%, with an average of 5.7% for Polska, 4.9% for NSF, 5.7% for Spain, 3.6% for Italia14, and 6.1% for EPAN, which are reasonable. Notice that the preselected logical network scenarios always show better resource efficiency. Also the results show that the second spine S_2 , in all networks, requires only a few extra resources more than

Table 6.4: Average and maximum logical link expected downtime.

Network	Scenario	Problem type	regular graph degree k	total no. of logical links	preselected logical links	Average/Maximum logical link expected downtime Class-1 (hrs/yr)			Average/Maximum logical link expected downtime Class-4 (hrs/yr)		
						S_1	S_2	S_3	S_1	S_2	S_3
Polska	1	duplicated logical network	3	36	no	0.42 / 1.00	0.47 / 1.30	0.45 / 1.10	62 / 142	65 / 250	69 / 173
	2		3	36	yes	0.34 / 1.20	0.35 / 1.50	0.33 / 1.10	40 / 138	56 / 321	40 / 183
	3		4	48	no	0.43 / 1.10	0.48 / 1.40	0.46 / 1.20	63 / 142	69 / 250	64 / 174
	4		4	48	yes	0.36 / 1.20	0.41 / 1.60	0.41 / 1.40	49 / 162	69 / 322	48 / 183
	5		5	60	no	0.41 / 1.10	0.45 / 1.50	0.44 / 1.20	64 / 135	76 / 276	64 / 207
	6		5	60	yes	0.39 / 1.20	0.46 / 1.70	0.41 / 1.10	59 / 162	76 / 333	58 / 204
	7	partitioned logical network	4	24	no	0.37 / 0.80	0.40 / 1.00	0.41 / 1.00	73 / 143	86 / 252	84 / 172
	8		4	24	yes	0.27 / 0.80	0.39 / 1.50	0.33 / 1.10	63 / 129	68 / 227	62 / 193
	9		5	30	no	0.36 / 1.00	0.44 / 1.30	0.33 / 1.10	72 / 116	71 / 202	62 / 193
	10		5	30	yes	0.39 / 1.20	0.53 / 1.70	0.41 / 1.50	58 / 119	55 / 192	70 / 182
	11		6	36	no	0.39 / 1.10	0.44 / 1.30	0.47 / 1.40	68 / 150	76 / 224	66 / 158
	12		6	36	yes	0.41 / 1.30	0.49 / 1.50	0.48 / 1.40	50 / 104	56 / 178	57 / 168
	13		7	42	no	0.40 / 1.10	0.47 / 1.50	0.47 / 1.40	70 / 142	77 / 228	73 / 173
	14		7	42	yes	0.42 / 1.20	0.52 / 1.60	0.46 / 1.30	63 / 153	61 / 182	69 / 168
NSF	1	duplicated logical network	3	42	no	0.48 / 1.17	0.50 / 1.81	0.46 / 0.82	82 / 217	70 / 214	59 / 181
	2		3	42	yes	0.38 / 1.24	0.35 / 1.49	0.33 / 0.85	46 / 195	54 / 246	42 / 181
	3		4	56	no	0.50 / 1.26	0.50 / 1.51	0.49 / 1.01	73 / 248	78 / 234	74 / 246
	4		4	56	yes	0.42 / 1.25	0.42 / 1.47	0.39 / 1.00	57 / 216	65 / 246	54 / 239
	5		5	70	no	0.49 / 1.23	0.51 / 1.54	0.49 / 1.04	73 / 234	80 / 265	72 / 245
	6		5	70	yes	0.41 / 1.12	0.46 / 1.73	0.42 / 1.04	58 / 202	66 / 260	59 / 228
	7	partitioned logical network	4	28	no	0.49 / 1.05	0.51 / 1.15	0.47 / 0.90	70 / 212	67 / 178	69 / 176
	8		4	28	yes	0.42 / 1.17	0.43 / 0.99	0.37 / 0.86	51 / 181	59 / 228	54 / 178
	9		5	35	no	0.51 / 1.18	0.54 / 1.28	0.47 / 0.96	63 / 200	65 / 228	70 / 179
	10		5	35	yes	0.46 / 1.12	0.50 / 1.45	0.44 / 1.02	42 / 145	53 / 212	42 / 127
	11		6	42	no	0.48 / 1.10	0.50 / 1.48	0.45 / 0.92	64 / 178	66 / 210	70 / 160
	12		6	42	yes	0.52 / 1.16	0.56 / 1.76	0.48 / 1.04	48 / 193	47 / 157	54 / 134
	13		7	49	no	0.52 / 1.19	0.54 / 1.79	0.51 / 1.02	69 / 162	66 / 178	65 / 167
	14		7	49	yes	0.50 / 1.19	0.56 / 2.07	0.49 / 1.02	48 / 176	53 / 174	51 / 162
Spain	1	duplicated logical network	3	42	no	0.49 / 1.34	0.52 / 1.53	0.57 / 1.29	73 / 190	76 / 221	86 / 236
	2		3	42	yes	0.29 / 1.09	0.39 / 1.48	0.34 / 1.25	42 / 166	49 / 233	42 / 184
	3		4	56	no	0.46 / 1.45	0.50 / 1.68	0.53 / 1.27	71 / 185	82 / 272	83 / 246
	4		4	56	yes	0.38 / 1.30	0.40 / 1.64	0.45 / 1.28	51 / 202	64 / 252	60 / 200
	5		5	70	no	0.46 / 1.36	0.52 / 1.62	0.56 / 1.32	68 / 174	83 / 284	79 / 243
	6		5	70	yes	0.41 / 1.52	0.43 / 1.65	0.49 / 1.33	66 / 191	75 / 284	64 / 231
	7	partitioned logical network	4	28	no	0.43 / 1.35	0.45 / 1.41	0.42 / 1.15	88 / 217	82 / 219	100 / 219
	8		4	28	yes	0.36 / 1.40	0.31 / 1.31	0.40 / 1.17	68 / 216	83 / 250	59 / 167
	9		5	35	no	0.82 / 1.50	0.41 / 1.30	0.45 / 1.07	130 / 183	81 / 234	85 / 188
	10		5	35	yes	0.38 / 1.10	0.51 / 1.51	0.51 / 1.26	58 / 153	49 / 162	55 / 150
	11		6	42	no	0.45 / 1.11	0.45 / 1.40	0.53 / 1.24	66 / 200	75 / 252	87 / 190
	12		6	42	yes	0.50 / 1.45	0.48 / 1.43	0.53 / 1.30	50 / 179	61 / 200	53 / 153
	13		7	49	no	0.49 / 1.28	0.53 / 1.65	0.56 / 1.28	60 / 162	75 / 231	67 / 164
	14		7	49	yes	0.47 / 1.31	0.53 / 1.54	0.58 / 1.28	48 / 134	55 / 165	60 / 153
Italia14	1	duplicated logical network	3	42	no	0.42 / 0.82	0.42 / 1.03	0.41 / 0.78	61 / 133	43 / 105	65 / 166
	2		3	42	yes	0.28 / 0.78	0.26 / 0.91	0.28 / 0.91	35 / 162	39 / 196	28 / 143
	3		4	56	no	0.43 / 0.93	0.39 / 0.84	0.41 / 0.88	65 / 163	52 / 174	58 / 147
	4		4	56	yes	0.31 / 0.81	0.32 / 0.92	0.31 / 0.78	40 / 160	36 / 178	41 / 147
	5		5	70	no	0.39 / 0.87	0.38 / 0.83	0.39 / 0.85	51 / 160	53 / 174	53 / 178
	6		5	70	yes	0.34 / 0.87	0.33 / 0.82	0.36 / 0.95	43 / 159	42 / 145	43 / 174
	7	partitioned logical network	4	28	no	0.35 / 0.76	0.37 / 0.80	0.35 / 0.71	81 / 179	70 / 197	87 / 164
	8		4	28	yes	0.26 / 0.71	0.18 / 0.68	0.21 / 0.60	59 / 162	66 / 160	59 / 129
	9		5	35	no	0.35 / 0.79	0.36 / 0.85	0.34 / 0.75	65 / 176	60 / 143	58 / 129
	10		5	35	yes	0.23 / 0.69	0.23 / 0.78	0.25 / 0.81	64 / 160	59 / 176	64 / 160
	11		6	42	no	0.37 / 0.83	0.38 / 0.86	0.37 / 0.82	60 / 131	51 / 145	61 / 119
	12		6	42	yes	0.30 / 0.85	0.28 / 0.94	0.33 / 0.77	56 / 159	79 / 176	65 / 145
	13		7	49	no	0.36 / 0.91	0.36 / 0.90	0.37 / 0.94	61 / 131	62 / 176	62 / 117
	14		7	49	yes	0.33 / 0.73	0.33 / 0.94	0.36 / 0.92	57 / 117	59 / 174	56 / 131
EPAN	1	duplicated logical network	3	48	no	0.62 / 1.67	0.71 / 2.14	0.71 / 1.51	83 / 198	100 / 308	111 / 267
	2		3	48	yes	0.45 / 1.57	0.43 / 1.96	0.42 / 1.42	39 / 178	50 / 221	49 / 231
	3		4	64	no	0.61 / 1.48	0.67 / 1.97	0.73 / 1.54	79 / 195	86 / 316	106 / 257
	4		4	64	yes	0.48 / 1.53	0.53 / 2.09	0.49 / 1.44	57 / 198	71 / 299	62 / 222
	5		5	80	no	0.60 / 1.74	0.70 / 2.26	0.73 / 1.56	78 / 214	97 / 335	105 / 269
	6		5	80	yes	0.53 / 1.47	0.56 / 2.23	0.56 / 1.61	60 / 205	71 / 267	75 / 262
	7	partitioned logical network	4	32	no	0.82 / 1.70	0.64 / 1.84	0.63 / 1.45	93 / 204	101 / 270	115 / 228
	8		4	32	yes	0.52 / 1.17	0.43 / 1.78	0.38 / 1.33	66 / 190	90 / 303	85 / 207
	9		5	40	no	0.57 / 1.61	0.63 / 1.95	0.63 / 1.39	81 / 188	106 / 316	110 / 236
	10		5	40	yes	0.57 / 1.49	0.64 / 2.23	0.56 / 1.48	68 / 188	59 / 204	71 / 202
	11		6	48	no	0.58 / 1.61	0.66 / 2.10	0.66 / 1.46	69 / 188	90 / 281	100 / 219
	12		6	48	yes	0.63 / 1.68	0.69 / 2.25	0.61 / 1.55	53 / 153	59 / 228	75 / 202
	13		7	56	no	0.59 / 1.63	0.66 / 2.07	0.70 / 1.56	76 / 185	82 / 253	98 / 202
	14		7	56	yes	0.61 / 1.79	0.71 / 2.13	0.65 / 1.54	55 / 183	66 / 284	67 / 215

Table 6.5: Average and maximum logical flows expected downtime.

Network	Scenario	Problem type	regular graph degree K	total no. of logical links	preselected logical links	Average/Maximum flow expected downtime Class-1 (hrs/yr)			Average/Maximum flow expected downtime Class-4 (hrs/yr)		
						S_1	S_2	S_3	S_1	S_2	S_3
Polska	1	duplicated logical network	3	36	no	0.78 / 1.90	0.87 / 2.30	0.85 / 2.00	118 / 287	123 / 432	133 / 385
	2		3	36	yes	0.53 / 1.50	0.43 / 1.70	0.44 / 1.40	54 / 177	42 / 333	42 / 215
	3		4	48	no	0.62 / 1.60	0.68 / 1.70	0.67 / 1.60	102 / 260	102 / 352	94 / 286
	4		4	48	yes	0.49 / 1.50	0.46 / 1.80	0.48 / 1.70	62 / 185	53 / 334	53 / 215
	5		5	60	no	0.50 / 1.30	0.53 / 1.50	0.52 / 1.40	89 / 230	99 / 372	85 / 260
	6		5	60	yes	0.46 / 1.30	0.47 / 1.70	0.46 / 1.30	71 / 200	67 / 333	66 / 264
	7	partitioned logical network	4	24	no	0.83 / 2.00	0.92 / 2.30	0.98 / 2.50	216 / 467	239 / 645	239 / 527
	8		4	24	yes	0.59 / 1.60	0.73 / 2.20	0.63 / 1.90	178 / 389	188 / 491	170 / 442
	9		5	30	no	0.59 / 1.50	0.68 / 1.70	0.66 / 1.70	180 / 403	174 / 421	161 / 388
	10		5	30	yes	0.56 / 1.60	0.71 / 2.00	0.57 / 1.80	150 / 341	117 / 368	151 / 409
	11		6	36	no	0.53 / 1.30	0.58 / 1.60	0.60 / 1.60	154 / 388	130 / 451	134 / 332
	12		6	36	yes	0.52 / 1.40	0.59 / 1.70	0.55 / 1.50	111 / 277	87 / 343	109 / 309
	13		7	42	no	0.50 / 1.30	0.53 / 1.60	0.54 / 1.60	122 / 283	116 / 385	135 / 343
	14		7	42	yes	0.48 / 1.20	0.56 / 1.70	0.49 / 1.50	114 / 276	90 / 306	122 / 355
NSF	1	duplicated logical network	3	42	no	1.02 / 2.47	1.07 / 2.83	0.98 / 2.22	164 / 474	155 / 511	126 / 370
	2		3	42	yes	0.91 / 2.70	0.60 / 2.29	0.72 / 1.97	139 / 468	91 / 439	90 / 371
	3		4	56	no	0.85 / 2.16	0.80 / 2.12	0.82 / 1.98	130 / 437	120 / 419	113 / 387
	4		4	56	yes	0.66 / 1.83	0.66 / 2.14	0.63 / 1.80	66 / 298	77 / 399	72 / 347
	5		5	70	no	0.69 / 1.71	0.68 / 2.03	0.70 / 1.86	101 / 338	114 / 414	102 / 399
	6		5	70	yes	0.64 / 1.62	0.62 / 2.33	0.61 / 1.88	92 / 272	72 / 362	75 / 317
	7	partitioned logical network	4	28	no	1.30 / 3.00	1.32 / 3.07	1.25 / 2.94	217 / 562	182 / 496	229 / 572
	8		4	28	yes	1.07 / 2.78	1.10 / 2.74	0.91 / 2.31	146 / 431	161 / 467	197 / 467
	9		5	35	no	0.95 / 2.17	1.01 / 2.43	0.92 / 2.12	179 / 435	179 / 506	182 / 452
	10		5	35	yes	0.85 / 2.01	0.88 / 2.62	0.83 / 2.25	106 / 302	122 / 449	121 / 317
	11		6	42	no	0.77 / 1.72	0.80 / 2.36	0.75 / 1.73	172 / 457	157 / 476	164 / 387
	12		6	42	yes	0.82 / 1.95	0.82 / 2.29	0.77 / 1.97	116 / 390	113 / 390	124 / 372
	13		7	49	no	0.75 / 1.92	0.73 / 2.34	0.76 / 1.70	157 / 422	148 / 407	151 / 409
	14		7	49	yes	0.72 / 1.92	0.73 / 2.63	0.73 / 1.80	108 / 355	98 / 375	112 / 353
Spain	1	duplicated logical network	3	42	no	0.98 / 2.36	1.00 / 2.64	1.16 / 2.58	150 / 417	151 / 420	183 / 490
	2		3	42	yes	0.48 / 1.96	0.60 / 2.33	0.69 / 2.45	83 / 340	97 / 386	115 / 425
	3		4	56	no	0.67 / 1.97	0.72 / 2.01	0.83 / 1.96	118 / 330	133 / 374	141 / 402
	4		4	56	yes	0.46 / 1.46	0.46 / 1.82	0.63 / 1.64	60 / 258	50 / 269	78 / 275
	5		5	70	no	0.60 / 1.64	0.63 / 1.87	0.73 / 1.90	103 / 294	118 / 386	116 / 355
	6		5	70	yes	0.47 / 1.61	0.50 / 1.88	0.61 / 1.58	62 / 236	70 / 321	71 / 250
	7	partitioned logical network	4	28	no	0.97 / 2.64	1.04 / 2.80	1.06 / 2.70	280 / 683	254 / 637	303 / 686
	8		4	28	yes	0.64 / 2.25	0.59 / 2.07	0.88 / 2.55	202 / 511	242 / 557	184 / 427
	9		5	35	no	0.76 / 1.81	0.72 / 2.01	0.81 / 1.95	183 / 444	188 / 459	216 / 474
	10		5	35	yes	0.53 / 1.48	0.75 / 2.09	0.81 / 2.06	136 / 351	115 / 332	139 / 343
	11		6	42	no	0.61 / 1.59	0.63 / 1.72	0.70 / 1.61	146 / 435	141 / 429	201 / 427
	12		6	42	yes	0.64 / 1.81	0.65 / 1.74	0.70 / 1.88	108 / 380	104 / 355	118 / 338
	13		7	49	no	0.59 / 1.53	0.61 / 1.69	0.68 / 1.53	123 / 316	129 / 409	148 / 340
	14		7	49	yes	0.57 / 1.43	0.62 / 1.78	0.66 / 1.77	95 / 294	100 / 342	132 / 337
Italia14	1	duplicated logical network	3	42	no	0.64 / 1.52	0.94 / 1.93	0.84 / 1.69	148 / 412	95 / 277	145 / 384
	2		3	42	yes	0.33 / 0.85	0.29 / 0.91	0.35 / 1.00	33 / 166	33 / 212	35 / 198
	3		4	56	no	0.70 / 1.40	0.66 / 1.46	0.67 / 1.50	125 / 319	112 / 333	130 / 353
	4		4	56	yes	0.32 / 0.86	0.31 / 0.93	0.37 / 1.05	36 / 197	28 / 178	41 / 214
	5		5	70	no	0.55 / 1.25	0.53 / 1.26	0.55 / 1.26	89 / 284	94 / 265	91 / 252
	6		5	70	yes	0.36 / 1.10	0.33 / 0.86	0.40 / 1.24	42 / 231	36 / 162	49 / 228
	7	partitioned logical network	4	28	no	0.95 / 2.04	0.96 / 1.96	0.94 / 1.97	248 / 559	208 / 475	264 / 588
	8		4	28	yes	0.62 / 1.47	0.44 / 1.17	0.53 / 1.27	173 / 434	207 / 505	173 / 430
	9		5	35	no	0.66 / 1.48	0.66 / 1.47	0.63 / 1.42	192 / 481	176 / 430	163 / 417
	10		5	35	yes	0.40 / 0.97	0.39 / 1.17	0.45 / 1.23	148 / 362	145 / 360	156 / 370
	11		6	42	no	0.54 / 1.18	0.58 / 1.28	0.54 / 1.29	166 / 360	144 / 410	159 / 395
	12		6	42	yes	0.39 / 1.04	0.34 / 1.03	0.44 / 1.10	134 / 316	170 / 387	161 / 369
	13		7	49	no	0.49 / 1.20	0.48 / 1.07	0.47 / 1.25	146 / 349	150 / 352	154 / 370
	14		7	49	yes	0.38 / 0.93	0.32 / 0.97	0.42 / 1.20	134 / 311	135 / 286	127 / 308
EPAN	1	duplicated logical network	3	48	no	1.31 / 3.12	1.46 / 3.94	1.53 / 3.60	185 / 462	217 / 586	229 / 608
	2		3	48	yes	0.99 / 3.84	1.19 / 4.32	1.30 / 3.77	64 / 261	155 / 470	141 / 435
	3		4	64	no	0.94 / 2.31	0.99 / 3.03	1.20 / 2.84	143 / 389	143 / 391	182 / 489
	4		4	64	yes	0.63 / 1.99	0.68 / 2.42	0.76 / 2.09	71 / 304	76 / 357	94 / 328
	5		5	80	no	0.84 / 2.13	0.87 / 2.59	1.02 / 2.46	118 / 340	145 / 466	156 / 474
	6		5	80	yes	0.62 / 1.78	0.65 / 2.34	0.77 / 1.98	68 / 267	79 / 323	100 / 304
	7	partitioned logical network	4	32	no	1.45 / 3.86	1.53 / 4.25	1.64 / 4.00	334 / 750	303 / 785	377 / 888
	8		4	32	yes	1.29 / 3.61	1.01 / 3.56	0.92 / 2.64	189 / 484	271 / 711	272 / 611
	9		5	40	no	1.05 / 2.83	1.07 / 3.11	1.19 / 2.71	225 / 583	267 / 760	298 / 670
	10		5	40	yes	0.95 / 2.45	0.92 / 2.60	0.91 / 2.44	163 / 449	158 / 459	206 / 493
	11		6	48	no	0.86 / 2.04	0.89 / 2.76	1.00 / 2.50	173 / 513	192 / 491	236 / 553
	12		6	48	yes	0.85 / 2.24	0.87 / 2.47	0.87 / 2.44	125 / 380	129 / 432	185 / 467
	13		7	56	no	0.73 / 2.05	0.79 / 2.27	0.89 / 2.30	176 / 441	160 / 437	240 / 506
	14		7	56	yes	0.75 / 1.99	0.82 / 2.40	0.83 / 2.17	113 / 380	109 / 393	156 / 416

Table 6.6: A comparison of the resource use between the spine and the baseline model.

Network	Scenario		regular graph degree k	total no. of logical links	preselected logical links	Resources required for link-disjoint path-pairs with No Spine	% Percentage of extra resource for link-disjoint path-pairs on the spine		
							S_1	S_2	S_3
Polska	1	3	36	no	994	12.6	1.4	5.3	
	2	3	36	yes	772/742/743	11.4	0.8	2.4	
	3	4	48	no	800	11.3	1.6	4.4	
	4	4	48	yes	684/672/674	11.8	1.0	3.4	
	5	5	60	no	667	12.1	1.2	4.6	
	6	5	60	yes	620/626/621	12.9	1.5	3.4	
NSF	1	3	42	no	1664	4.8	3.1	7.0	
	2	3	42	yes	1342/1365/1371	4.7	1.3	6.2	
	3	4	56	no	1343	5.1	1.7	7.1	
	4	4	56	yes	1178/1173/1166	5.3	1.6	8.6	
	5	5	70	no	1167	6.0	2.5	8.2	
	6	5	70	yes	1086/1069/1095	7.1	2.0	6.6	
Spain	1	3	42	no	1560	6.3	3.4	13.1	
	2	3	42	yes	1098/1105/1136	2.7	2.1	8.3	
	3	4	56	no	1172	5.6	3.0	10.1	
	4	4	56	yes	1007/1028/1048	4.2	1.4	7.0	
	5	5	70	no	1064	6.4	2.4	10.2	
	6	5	70	yes	959/948/949	4.2	2.4	10.1	
Italia14	1	3	42	no	1309	4.4	7.5	5.8	
	2	3	42	yes	814/775/855	1.7	0.4	1.6	
	3	4	56	no	990	7.4	2.2	6.1	
	4	4	56	yes	778/763/799	2.4	0.6	3.3	
	5	5	70	no	870	5.1	1.8	6.3	
	6	5	70	yes	755/730/757	2.6	1.3	4.8	
EPAN	1	3	48	no	2585	5.2	2.3	10.9	
	2	3	48	yes	1853/1790/1757	2.0	0.9	11.1	
	3	4	64	no	1980	5.2	2.1	13.2	
	4	4	64	yes	1587/1606/1598	5.5	1.5	9.6	
	5	5	80	no	1760	7.1	1.8	13.0	
	6	5	80	yes	1492/1497/1475	5.4	1.6	11.5	

the baseline model. This confirms that the design of the spine as well as the logical network layout are of high significance to the whole model. Second, in terms of availability levels offered by the network, Table 6.7 shows the average and maximum expected downtimes for both class-1 and class-4 logical flows in the no-spine model. By comparing these results to the results of the spine model in Table 6.5, we observe that on average the expected downtimes for class-1 of the no-spine model are 3 times larger than the corresponding results of the spine model for the Polska, Spain and EPAN networks, 2 times for the NSF, and 4 times for Italia14. Class-4 downtimes in Table 6.7 are fairly similar to the spine model (Table 6.5) in all networks except Italia14 and Polska networks where it is on average 2 and 1.6 times,

Table 6.7: Average and maximum flow expected downtime for the baseline model.

Network	Scenario	regular graph degree k	total no. of logical links	preselected logical links	Average/Maximum flow expected downtime Class-1 (hrs/yr)			Average/Maximum flow expected downtime Class-4 (hrs/yr)		
					S_1	S_2	S_3	S_1	S_2	S_3
Polska	1	3	36	no	2.43 / 5.34	2.43 / 5.34	2.43 / 5.34	163 / 310	163 / 310	163 / 310
	2	3	36	yes	1.59 / 3.68	1.44 / 3.95	1.39 / 3.46	104 / 184	100 / 190	104 / 212
	3	4	48	no	1.92 / 4.41	1.92 / 4.41	1.92 / 4.41	130 / 228	130 / 228	130 / 228
	4	4	48	yes	1.46 / 3.31	1.42 / 3.95	1.42 / 3.63	98 / 168	95 / 168	97 / 173
	5	5	60	no	1.44 / 3.84	1.44 / 3.84	1.44 / 3.84	124 / 337	124 / 337	124 / 337
	6	5	60	yes	1.36 / 4.94	1.29 / 3.47	1.29 / 3.42	105 / 283	121 / 338	121 / 332
NSF	1	3	36	no	2.36 / 4.97	2.36 / 4.97	2.36 / 4.97	154 / 305	154 / 305	154 / 305
	2	3	36	yes	1.60 / 3.46	1.58 / 3.74	1.57 / 3.54	95 / 187	92 / 180	95 / 180
	3	4	48	no	1.92 / 3.69	1.92 / 3.69	1.92 / 3.69	125 / 207	125 / 207	125 / 207
	4	4	48	yes	1.47 / 3.07	1.47 / 3.23	1.43 / 3.31	89 / 174	87 / 167	87 / 160
	5	5	60	no	1.62 / 3.30	1.62 / 3.30	1.62 / 3.30	107 / 193	107 / 193	107 / 193
	6	5	60	yes	1.37 / 2.97	1.40 / 3.05	1.42 / 2.97	83 / 147	85 / 160	87 / 160
Spain	1	3	36	no	3.22 / 8.01	3.22 / 8.01	3.22 / 8.01	196 / 378	196 / 378	196 / 378
	2	3	36	yes	1.77 / 6.16	1.68 / 4.97	1.76 / 4.48	108 / 203	107 / 219	114 / 219
	3	4	48	no	2.35 / 6.75	2.35 / 6.75	2.35 / 6.75	142 / 267	142 / 267	142 / 267
	4	4	48	yes	1.73 / 5.29	1.68 / 4.88	1.85 / 4.84	106 / 203	105 / 203	110 / 203
	5	5	60	no	2.03 / 4.69	2.03 / 4.69	2.03 / 4.69	131 / 227	131 / 227	131 / 227
	6	5	60	yes	1.69 / 4.87	1.70 / 4.73	1.67 / 4.61	102 / 203	103 / 203	105 / 203
Italia14	1	3	36	no	3.63 / 8.73	3.63 / 8.73	3.63 / 8.73	215 / 419	215 / 419	215 / 419
	2	3	36	yes	1.53 / 5.13	1.40 / 4.11	1.62 / 4.18	119 / 197	112 / 156	123 / 207
	3	4	48	no	2.61 / 6.83	2.61 / 6.83	2.61 / 6.83	158 / 289	158 / 289	158 / 289
	4	4	48	yes	1.53 / 4.35	1.50 / 4.78	1.64 / 4.60	115 / 166	112 / 156	118 / 207
	5	5	60	no	2.27 / 6.18	2.27 / 6.18	2.27 / 6.18	139 / 248	139 / 248	139 / 248
	6	5	60	yes	1.58 / 4.30	1.54 / 6.56	1.64 / 4.85	114 / 197	109 / 156	115 / 187
EPAN	1	3	36	no	8.21 / 20.11	8.21 / 20.11	8.21 / 20.11	294 / 605	294 / 605	294 / 605
	2	3	36	yes	2.25 / 7.81	1.87 / 5.68	1.78 / 4.46	126 / 250	113 / 222	119 / 215
	3	4	48	no	2.64 / 6.56	2.64 / 6.56	2.64 / 6.56	162 / 306	162 / 306	162 / 306
	4	4	48	yes	1.78 / 5.32	1.87 / 5.02	1.81 / 5.43	105 / 215	107 / 215	114 / 222
	5	5	60	no	2.31 / 5.40	2.31 / 5.40	2.31 / 5.40	147 / 271	147 / 271	147 / 271
	6	5	60	yes	1.83 / 4.91	1.81 / 5.58	1.71 / 4.90	106 / 215	106 / 215	108 / 215

respectively, the downtime of the spine model. The no-spine model, however, achieves lower maximum flow downtimes with respect to class-4. In general, the results in Tables 6.6 and 6.7 show that the spine model stretches the upper bound of the availability of the 1+1 protection to offer a higher availability level with a fairly small increase in resource use.

Note that the utility of the spine depends on the cost of designing the spine and the traffic load. In particular, the percentage of class 1 traffic affects the amount of extra resources needed, here the percentage is 50%, in reality we would expect it to be a much smaller percentage (*e.g.*, $\leq 5\%$). For this, we repeat some of the aforementioned scenarios and vary

Table 6.8: Traffic ratio scenarios.

# scenario	Traffic ratio	Class-1 demand	Class-4 demand
d1	50/50	1	1
d2	25/75	0.5	1.5
d3	20/80	0.3	1.7
d4	10/90	0.2	1.8
d5	05/95	0.1	1.9

the ratio of the traffic between the two classes. We consider four additional scenarios besides the previous scenario (50/50). Table 6.8 shows the tested scenarios and the corresponding class-1/class-4 traffic ratio and the unit of demand for each class. Note that the total sum of all the scenarios is kept the same (full mesh, 2 units of demand). We compare the percentage of increase in resources usage compared to the no spine case (*i.e.*, the baseline model) for the Polska network case. The results are shown in Figure 6.6. One can observe that in almost all scenarios the spine model requires less additional resources compared to the no spine model as the percentage of class-1 traffic decreases. This, however, comes at the expense of increasing class expected downtimes, especially in the partitioned problem. Technically, as class-4 traffic increases it becomes the dominant part of the objective function (6.5). Hence the algorithm, in order to minimize the total resources, assigns shorter logical paths to class-4 flows. To overcome this limitation, we can add weights to the objective function to control the dominance of either class.

6.4 PROTECTION AT UPPER LAYER

The multilayer spine model introduced in Section 6.3 requires each class of traffic to maintain a separate logical network. Hence the number of logical subnetworks is proportional to the number of classes. This is not only because the mapping into or off the spine is projected toward the physical layer, but also because the differentiation in protection occurs at the physical layer. However, providing protection at the upper layer changes the proportional relation between the number of logical subnetworks and classes.

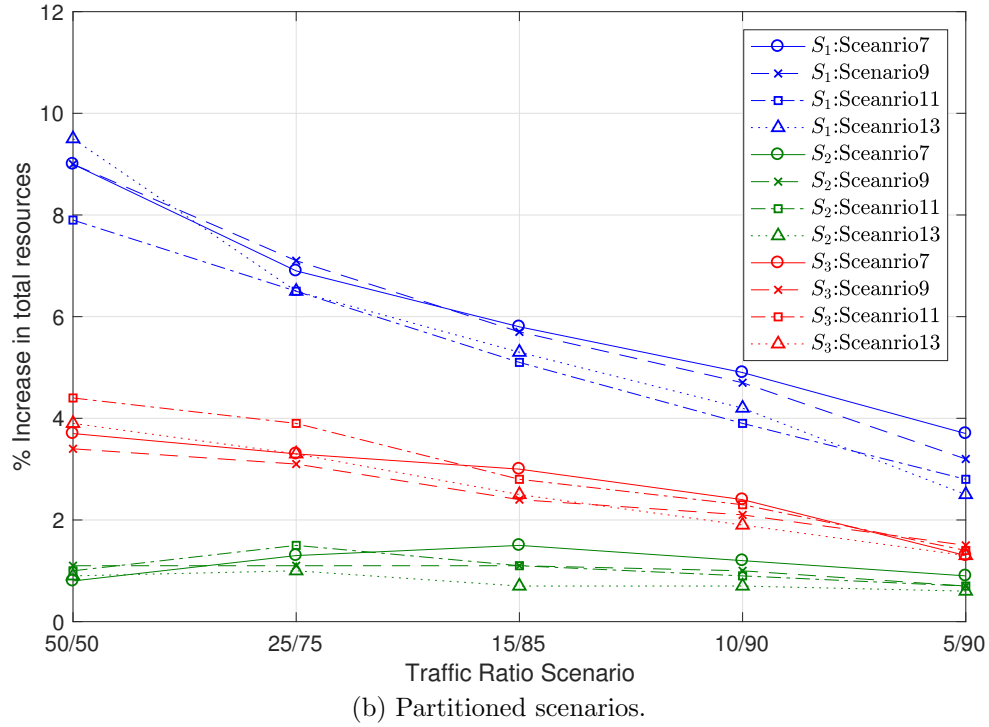
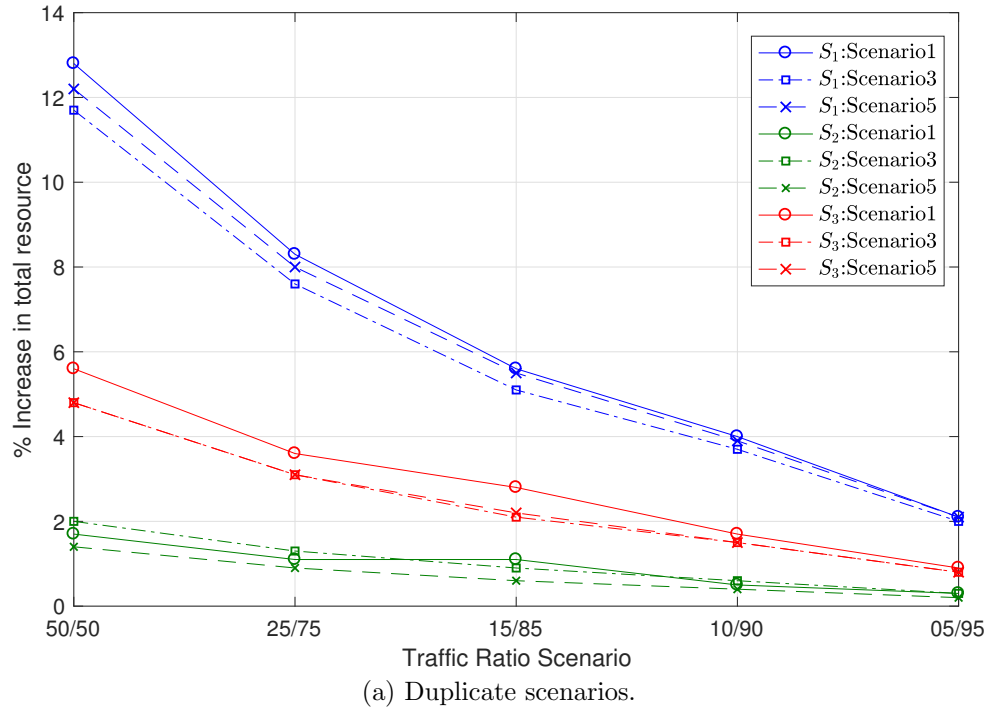


Figure 6.6: Percentage of increase in total resources compared to the baseline model for the Polska network.

Here, we consider a two layer network as in Figure 6.2, where \mathcal{G}_P , \mathcal{G}_L , \mathcal{G}_S , and D^ϕ are given. Also we consider two classes of flows: flows of one class require high availability levels while flows of the other class are of a lesser importance and have no strict availability requirements *i.e.*, class 1 and 4 in Table 6.1, respectively. Flows of the high availability class are routed through a fully link-disjoint working and backup logical path-pair. One of either paths is required to be routed through logical links that are mapped entirely to the physical layer spine. The other logical path is mapped to an unrestricted underlay path that is disjoint from the path on the spine. None of the physical paths are protected. Flows of the lower availability class are routed freely on the network through a logical path mapped to an unprotected physical path. Therefore, one can distinguish two logical networks based on their mapping. Further, the two logical networks can be used to route additional classes, *i.e.*, class 2 and 3 in Table 6.1. Flows of Class-2 can be routed through a fully or partially link-disjoint logical path-pair that are mapped freely on the network (*i.e.*, on the same logical network of class-4). Class-3 flows are routed on the same logical network of class-1 but with no protection. Class-2 and class-3 flows are expected to have availability less than class-1 and higher than class-4 flows.

6.4.1 Notation

In addition to the notation described in Section 6.3.1, we define the following:

Availability:

a_{ij} availability of physical link (i, j) .

A_{st} availability of logical link (s, t) :

$$A_{st} = \prod_{ij|st \rightarrow ij} a_{ij} \quad (6.18)$$

A_{mn}^{WP} (A_{mn}^{BP}) availability of flow mn working path (backup path).

$$A_{mn}^{WP} = \prod_{st|WP_{mn} \rightarrow st} A_{st} \quad (6.19)$$

A_{mn} assuming WP_{mn} and BP_{mn} are fully edge-disjoint:

$$A_{mn} = 1 - (1 - A_{mn}^{WP}) (1 - A_{mn}^{BP}) \quad (6.20)$$

EDT Expected downtime in hours per year. Given availability A :

$$EDT_{hrs/yr} = 24 \times 365 \times (1 - A) \quad (6.21)$$

Variables:

$Zx_{st}^{d_{\phi}^{mn}} (Zy_{st}^{d_{\phi}^{mn}})$	a continuous variable denoting the amount of class ϕ demand mn 's WP (BP) routed through logical link (s, t) .
$X_{ij}^{st} (Y_{ij}^{st})$	a binary variable indicating whether logical link (s, t) on logical network X (Y) is mapped to physical link (i, j) .
$B_x^{mn, \phi}_{st, ij} (B_y^{mn, \phi}_{st, ij})$	a continuous variable denoting the resources provisioned on physical link (i, j) to serve class- ϕ demand mn 's WP (BP) that is routed through logical link (s, t) on logical network X (Y).
$x_{st} (y_{st})$	a binary variable indicates whether a logical link (s, t) is selected in logical network X (Y).
$\chi_{ij}^{mn} (\psi_{ij}^{mn})$	a binary variable indicates whether demand mn uses physical link (i, j) as a part of its crosslayer WP (BP).

6.4.2 Problem Statement

Given a network, we create two logical subnetworks, named X and Y . Links of the logical subnetwork X are mapped onto the physical spine to create a logical spine with high availability, whereas logical subnetwork Y links are mapped unrestricted on the network. This can be done in one of two ways: duplicate logical links or a partitioned logical network. In the first approach, we assume that each logical link is duplicated and a similar set of links would be part of the logical subnetwork X and Y . Although both subnetworks have the same set of links, similar links are likely to have different mapping and bandwidth. In the second method, the logical network is partitioned into two subnetworks X and Y , and the logical links are split into two non-overlapped sets.

It is expected that subnetwork X will have higher availability than Y as it will be mapped entirely on the spine. Thus subnetwork X can be set up to carry the working path for flows that require high availability (*i.e.*, class-1) while subnetwork Y carries the backup paths of these flows. It is required that the path-pair for each flow is fully link disjoint which means the WP on X and the BP on Y share no common physical link. Note that the disjointness requirement might lengthen these paths, so it is more appropriate to find shorter paths on each network if disjointness is not required. Hence, after creating the logical subnetworks, flows with lower availability requirements (*i.e.*, class-3 & 4 in Table 6.1) can be routed on X or Y by performing weighted routing.

Here we propose a Mixed Integer Linear Programming (MILP) that finds the logical subnetworks X and Y and performs appropriate mapping to have differentiated logical routes for upper layer flows. Further, we propose a Linear Programming (LP) formulation for routing lower classes flows on these subnetworks.

6.4.3 Problem Formulation

6.4.3.1 Logical Subnetworks Design Problem Formulation We formulate the problem of determining subnetworks X and Y as a Mixed Integer Linear Programming (MILP) problem given by (6.22)-(6.46). This formulation can be used for both the duplicate logical links or the partitioned logical network problem with a minor addition for the latter. The aim is to create two logical subnetworks mapped in a way such that they have different levels of availability and support a fully link-disjoint path-pair between any two communication nodes. The objective of the MILP problem (6.22) is to minimize the total resources reserved in the physical links (*i.e.*, provisioned capacity) for routing and mapping all upper layer flows and links. This includes both working and backup resources. Most of the constraints are shown here in pairs; one for X and the other for Y . The main difference is the mapping scope *i.e.*, the spine or the entire network.

$$\begin{aligned}
& \underset{\substack{B_x^{mn,\phi} \\ B_y^{mn,\phi} \\ B_x^{st} \\ B_y^{st} \\ B_x^{ij} \\ B_y^{ij}}}{\text{minimize}} & \sum_{\phi} \sum_{mn} \sum_{st \in \mathcal{E}_L} \left(\sum_{ij \in \mathcal{E}_S} B_x^{mn,\phi} B_x^{st} + \sum_{ij \in \mathcal{E}_P} B_y^{mn,\phi} B_y^{st} \right) \\
& \text{s.t.}
\end{aligned} \tag{6.22}$$

$$\sum_{kt \in \mathcal{E}_L} Zx_{kt}^{d_\phi^{mn}} - \sum_{sk \in \mathcal{E}_L} Zx_{sk}^{d_\phi^{mn}} = \begin{cases} d_\phi^{mn} & \text{if } k = m \\ -d_\phi^{mn} & \text{if } k = n \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{V}_L, d_\phi^{mn}, \phi = 1 \quad (6.23)$$

$$\sum_{kt \in \mathcal{E}_L} Zy_{kt}^{d_\phi^{mn}} - \sum_{sk \in \mathcal{E}_L} Zy_{sk}^{d_\phi^{mn}} = \begin{cases} d_\phi^{mn} & \text{if } k = m \\ -d_\phi^{mn} & \text{if } k = n \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{V}_L, d_\phi^{mn}, \phi = 1 \quad (6.24)$$

$$\sum_{mn} Zx_{st}^{d_\phi^{mn}} \leq Mx_{st} \quad \sum_{mn} Zx_{st}^{d_\phi^{mn}} \geq x_{st} \quad \forall (s, t) \in \mathcal{E}_L, \phi = 1 \quad (6.25)$$

$$\sum_{mn} Zy_{st}^{d_\phi^{mn}} \leq My_{st} \quad \sum_{mn} Zy_{st}^{d_\phi^{mn}} \geq y_{st} \quad \forall (s, t) \in \mathcal{E}_L, \phi = 1 \quad (6.26)$$

$$\sum_{kj \in \mathcal{E}_S} X_{kj}^{st} - \sum_{ik \in \mathcal{E}_S} X_{ik}^{st} = \begin{cases} x_{st} & \text{if } k = s \\ -x_{st} & \text{if } k = t \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{V}_S \quad (6.27)$$

$$\sum_{kj \in \mathcal{E}_P} Y_{kj}^{st} - \sum_{ik \in \mathcal{E}_P} Y_{ik}^{st} = \begin{cases} y_{st} & \text{if } k = s \\ -y_{st} & \text{if } k = t \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{V}_P \quad (6.28)$$

$$X_{ij}^{st} + X_{ji}^{st} \leq 1 \quad \forall (i, j) \in \mathcal{E}_S, (s, t) \in \mathcal{E}_L \quad (6.29)$$

$$Y_{ij}^{st} + Y_{ji}^{st} \leq 1 \quad \forall (i, j) \in \mathcal{E}_P, (s, t) \in \mathcal{E}_L \quad (6.30)$$

$$\sum_{ij \in \mathcal{E}_S} X_{ij}^{st} + \sum_{hi \in \mathcal{E}_S} X_{hi}^{st} \leq 2 \quad \forall i \in \mathcal{V}_S, (s, t) \in \mathcal{E}_L \quad (6.31)$$

$$\sum_{ij \in \mathcal{E}_P} Y_{ij}^{st} + \sum_{hi \in \mathcal{E}_P} Y_{hi}^{st} \leq 2 \quad \forall i \in \mathcal{V}_P, (s, t) \in \mathcal{E}_L \quad (6.32)$$

$$\sum_{kj \in \mathcal{E}_S} B_x^{mn, \phi}_{kj} - \sum_{ik \in \mathcal{E}_S} B_x^{mn, \phi}_{ik} = \begin{cases} Zx_{st}^{d_\phi^{mn}} & \text{if } k = s \\ -Zx_{st}^{d_\phi^{mn}} & \text{if } k = t \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{V}_S, \phi = 1 \quad (6.33)$$

$$\sum_{kj \in \mathcal{E}_L} B_y^{mn, \phi}_{kj} - \sum_{ik \in \mathcal{E}_L} B_y^{mn, \phi}_{ik} = \begin{cases} Zy_{st}^{d_\phi^{mn}} & \text{if } k = s \\ -Zy_{st}^{d_\phi^{mn}} & \text{if } k = t \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{V}_P, \phi = 1 \quad (6.34)$$

$$B_x^{mn,\phi}_{st} \leq M X_{ij}^{st} \quad \forall (i,j) \in \mathcal{E}_S, (s,t) \in \mathcal{E}_L, d_\phi^{mn} \quad (6.35)$$

$$B_y^{mn,\phi}_{st} \leq M Y_{ij}^{st} \quad \forall (i,j) \in \mathcal{E}_P, (s,t) \in \mathcal{E}_L, d_\phi^{mn} \quad (6.36)$$

$$\sum_{mn} B_x^{mn,\phi}_{st} \geq X_{ij}^{st} \quad \forall (i,j) \in \mathcal{E}_S, (s,t) \in \mathcal{E}_L \quad (6.37)$$

$$\sum_{mn} B_y^{mn,\phi}_{st} \geq Y_{ij}^{st} \quad \forall (i,j) \in \mathcal{E}_P, (s,t) \in \mathcal{E}_L \quad (6.38)$$

$$\sum_{st \in \mathcal{E}_L} B_x^{mn,\phi}_{st} \leq M \chi_{ij}^{mn} \quad \forall d_\phi^{mn}, (s,t) \in \mathcal{E}_L, (i,j) \in \mathcal{E}_S, \phi = 1 \quad (6.39)$$

$$\sum_{st \in \mathcal{E}_L} B_y^{mn,\phi}_{st} \leq M \psi_{ij}^{mn} \quad \forall d_\phi^{mn}, (s,t) \in \mathcal{E}_L, (i,j) \in \mathcal{E}_P, \phi = 1 \quad (6.40)$$

$$\chi_{ij}^{mn} + \psi_{ij}^{mn} \leq 1 \quad \forall d_\phi^{mn}, (i,j) \in \mathcal{E}_S \quad (6.41)$$

$$\sum_{\phi} \sum_{st \in \mathcal{E}_L} (B_x^{mn,\phi}_{st} + B_y^{mn,\phi}_{st}) \leq c_{ij} \quad \forall (i,j) \in \mathcal{E}_P \quad (6.42)$$

$$\sum_{\phi} \sum_{mn} (Zx_{st}^{d_\phi^{mn}} + Zy_{st}^{d_\phi^{mn}}) \leq W_{st} \quad \forall (s,t) \in \mathcal{E}_L \quad (6.43)$$

$$x_{st} + y_{st} \leq 1 \quad \forall (s,t) \in \mathcal{E}_L \quad (6.44)$$

$$X_{ij}^{st}, Y_{ij}^{st}, x_{s,t}, y_{s,t}, \chi_{ij}^{mn}, \psi_{ij}^{mn} \quad \text{binary} \quad (6.45)$$

$$Zx_{st}^{d_\phi^{mn}}, Zy_{st}^{d_\phi^{mn}}, B_x^{mn,\phi}_{st}, B_y^{mn,\phi}_{st} \geq 0 \quad (6.46)$$

The formulation adapts multiple sets of flow conservation constraints shown by constraint sets (6.23),(6.24),(6.27),(6.28),(6.33), and(6.34) to find eligible paths in both layers. First, constraint sets (6.23) and (6.24) route logical demands (flows) of QoR class-1 ($\phi = 1$) through working paths and backup paths on the logical layer, respectively. Constraints (6.25) and (6.26) translate the routing variables into binary variables and determine whether a logical link is used. The next two sets perform the restricted crosslayer mapping for logical links using binary variables X_{ij}^{st} and Y_{ij}^{st} . A logical link is mapped only if it is used. The mapping can be on-spine as in constraint set (6.27) or freely on the network as in (6.28). Constraint sets (6.29)-(6.32) ensure the mapping is loop free *i.e.*, forbid revisiting

a link or a node in the physical path. The last flow conservation constraint sets (6.33) and (6.34) perform the crosslayer routing and mapping for each flow. Constraints (6.35) (6.36), (6.37) (6.38) relate the per-flow crosslayer routing/mapping variables $B_x^{mn,\phi}_{st\ ij}$ and $B_y^{mn,\phi}_{st\ ij}$ to the binary logical link mapping variables X_{ij}^{st} and Y_{ij}^{st} , respectively. In other words, these constraints ensure the per-flow mapping is consistent with the link mapping, such that, each logical link is mapped onto one path only. The big- M here is a constant set to $M = \sum_{mn} d_{\phi=1}^{mn}$. Then, constraint sets (6.39) and (6.40) transfer these per-flow variables $B_x^{mn,\phi}_{st\ ij}$ and $B_y^{mn,\phi}_{st\ ij}$ into binary variables χ_{ij}^{mn} and ψ_{ij}^{mn} . It forces the binary variable to one if flow mn is routed through any logical link (s, t) that is mapped to the physical link (i, j) . Further for each protected flow mn , constraint (6.41) ensures that logical links carrying resources for a particular logical path-pair are mapped to diverse lower layer paths. The big- M in these constraints is set to $M = 10 \times \sum_{mn} d_{\phi=1}^{mn}$. Constraint sets (6.42) and (6.43) are the capacity constraints of the physical and logical links, respectively. For the partitioned logical network problem, constraint set (6.44) is added to separate the logical spine X from the other logical subnetwork. Lastly, constraints (6.45) and (6.46) declare the variables.

6.4.3.2 Routing on Logical Subnetworks Problem Once we create the two logical subnetworks, we can route logical demands for lower QoR classes (*i.e.*, class 3 and 4 in Table 6.1). From the obtained results, we use the binary mapping variables X_{ij}^{st} and Y_{ij}^{st} to compute weights $w(x_{st})$ and $w(y_{st})$ for logical links on subnetwork X and Y , respectively, as follows:

$$w(x_{st}) = \sum_{ij \in \mathcal{E}_S} X_{ij}^{st} \quad (6.47)$$

$$w(y_{st}) = \sum_{ij \in \mathcal{E}_P} Y_{ij}^{st} \quad (6.48)$$

These weights represent the capacities needed in the lower layer for a logical link to carry a single unit of flow. Now, let $d_{\phi}^{mn} \in D_{\phi}$ be the upper layer demands of QoR class ϕ between node pair mn . Similarly, $Zx_{st}^{d_{\phi}^{mn}}$ ($Zy_{st}^{d_{\phi}^{mn}}$) takes the same definition as in the previous formulation. Then, we solve the LP problem below to find the minimum weight routes for demands on each subnetwork:

$$\begin{aligned}
& \underset{\substack{Zx_{st}^{d_{\phi=3}^{mn}} \\ Zy_{st}^{d_{\phi=4}^{mn}}}}{\text{minimize}} & \sum_{mn} \sum_{st \in \mathcal{E}_L} \left[w(x_{st}) \cdot Zx_{st}^{d_{\phi=3}^{mn}} + w(y_{st}) \cdot Zy_{st}^{d_{\phi=4}^{mn}} \right] \\
& \text{s.t.}
\end{aligned} \tag{6.49}$$

$$\sum_{kt \in \mathcal{E}_L} x_{kt} \cdot Zx_{kt}^{d_{\phi}^{mn}} - \sum_{sk \in \mathcal{E}_L} x_{sk} \cdot Zx_{sk}^{d_{\phi}^{mn}} = \begin{cases} d_{\phi}^{mn} & \text{if } k = m \\ -d_{\phi}^{mn} & \text{if } k = n \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{V}_L, d_{\phi}^{mn}, \phi = 3 \tag{6.50}$$

$$\sum_{kt \in \mathcal{E}_L} y_{kt} \cdot Zy_{kt}^{d_{\phi}^{mn}} - \sum_{sk \in \mathcal{E}_L} y_{sk} \cdot Zy_{sk}^{d_{\phi}^{mn}} = \begin{cases} d_{\phi}^{mn} & \text{if } k = m \\ -d_{\phi}^{mn} & \text{if } k = n \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{V}_L, d_{\phi}^{mn}, \phi = 4 \tag{6.51}$$

Although for a class-1 demand it is ensured (by the MILP formulation) that the WP in subnetwork-X is physically disjoint with the BP path in subnetwork-Y, the LP allows a path (in subnetwork-X) for a class-3 demand to share physical links with a path (in subnetwork-Y) for a class-4 demand.

6.4.4 Computational Complexity

The logical subnetwork design problem of subsection 6.4.3.1 is NP-complete since the problem of finding crosslayer disjoint path-pairs in multilayer networks was shown to be NP-complete [132, 133]. This suggests that solving the MILP in (6.22)–(6.46) for optimal solution is only possible for small instances and might incur excessive time for large networks, or be intractable. Also note that the partitioned logical network version of the proposed problem becomes even more complex after adding constraint set (6.44) (*i.e.*, an over-constrained problem [134]). Our numerical results confirm this, as we were able to solve the proposed MILP for a number of networks using the Gurobi solver (with the built-in ILP relaxation methods). The execution time was, roughly, on the order of tens to hundreds of seconds for the duplicate logical network problem and tens of minutes to a few hours for the partitioned version of the problem. The routing problem of subsection 6.4.3.2 is a linear programming problem (LP) which can be solved in polynomial time bound fashion using interior point method or the simplex method. Here the execution time was on the order of a few milliseconds.

6.4.5 Numerical Study

6.4.5.1 Scenarios We consider the same set of scenarios used in Section 6.3.4.1. For each network and spine, we ran 14 scenarios as shown on the left side of Table 6.9 where S_1, S_2 and S_3 refer to the spine number. These scenarios vary in a number of criteria: node degree of the k -regular graph, whether the graph is completely random or not, and the problem type, duplicate or partitioned logical network. Each scenario is repeated 7 times and the results are averaged. Further, we consider a full mesh of upper layer traffic flows with a single unit bandwidth demand for each traffic class ($d_\phi^{mn} = 1, \forall \phi, mn$). An upper layer flow can be routed through a single path on X (QoR class-3), a single path on Y (QoR class-4), or a disjoint path-pair through X and Y (QoR class-1). A path-pair on Y only (QoR class-2) is not included here.

We compare the averaged results (*i.e.*, over the 7 repetitions) in terms of resource use, logical link availability per subnetwork, and end-to-end flow availability per path type. Resource use is the amount of physical capacity required to realize the logical links aggregate bandwidth. This is given by the objective function (6.22) in the optimization problem. We assume unlimited bandwidth capacity at both the physical and logical layers, effectively relaxing constraints (6.42) and (6.43). The mapping and routing information are used to compute logical link availability a_{st} and logical flow availability a_{mn} for each link and flow using equations (6.18)-(6.20). This is computed for logical subnetworks X and Y and for each path type. After computing individual link and flow availabilities, we use the average values $\bar{A}_{st} = \sum_{st \in \mathcal{E}_L} A_{st} / |\mathcal{E}_L|$ and $\bar{A}_{mn} = \sum_{mn \in \mathcal{D}} A_{mn} / |\mathcal{V}_L| \times (|\mathcal{V}_L| - 1) / 2$ as metrics for comparison. Additionally we compare the minimum link availability $A_{st}^{\min} = \min_{st \in \mathcal{E}_L} A_{st}$ and minimum flow availability $A_{mn}^{\min} = \min_{mn \in \mathcal{D}} A_{mn}$ which are used to determine the expected downtime in hours per year as in equation (6.4).

6.4.5.2 Results We use AMPL/Gurobi on NEOS server [129–131] to solve our models. The results are with 0% integrality gap for the duplicate scenarios and within a 3% integrality gap for the partitioned logical network scenarios, and throughout our analysis we compare the results to the results obtained in Section 6.3, for the model which provides protection at

the bottom layer utilizing the spine.

Resource Use for QoR class-1 Traffic: The main objective of our design is to minimize the total physical resources used for realizing logical links of both subnetworks. Table 6.9 shows the resources used by path type for all scenarios considering only QoR class-1 flows with demand $d_\phi^{mn} = 1, \forall mn, \phi = 1$. Note that table entries “n/a” means that the model is infeasible. The reason for such a case is that the layout and density of the logical network is insufficient to satisfy the disjointness constraint in the lower layer.

Examining the results for the Polska network, we can make a number of observations as follows. First, resource use decreases as the logical network gets denser. For example, scenario 6 with the largest number of logical links yields the smallest resource use. Second, there is a 7.1, 17.1, and 16.8% decrease in resources for spine S_1 , S_2 , and S_3 respectively, when spine links are preselected in the logical network. A third observation can be made comparing the resource utilization between the two problem types (*i.e.*, duplicate versus partitioned). In order to compare the two types properly, we consider similar scenarios with comparable number of logical links: specifically scenarios 1 and 2 with 36 logical links to scenarios 11 and 12, respectively. Similarly, 3 and 4 to 13 and 14. When the logical network is completely random (*e.g.*, scenarios 1 & 3), 17.6–35.2% more resources are required for the problem with duplicate links. In contrast, this percentage becomes at most 17% when some links are preselected (*e.g.*, scenarios 2 & 4). Fourth, the contribution of the working paths (paths on X) to the total resources needed for the path-pair (QoR class-1) is less than the backup paths (paths on Y) contribution. For example, for the Polska network the contribution of the WP to the total resources (averaged over the 14 scenarios) is 47.6, 43.5, 45.3% for S_1 , S_2 , and S_3 respectively, compared to 52.4, 56.5, 54.7% for the BP contribution. Fifth, S_2 (*i.e.*, the spine with maximum path availability on the spine) always requires lower resources. Further, spines with shorter diameters (see Table 6.2) tend to require less resources, *e.g.*, for the Polska network, in 12 of the 14 scenarios, S_3 requires less resources than S_1 and is the second best regarding total resource usage. Moreover, the results for the other networks (NSF, Spain, Italia14, EPAN) are consistent with the observations above.

Note that our observations are similar to the ones obtained in Section 6.3.4.2 with protection given at the lower layer. However, configuring protection at the upper layer yields a

Table 6.9: QoR class-1 average resource use for the three spines in all scenarios.

Network	Scenario	Problem type	regular graph degree k	total no. of logical links	preselected logical links	Average total physical resources			Average subnetwork-X resources			Average subnetwork-Y resources		
						S_1	S_2	S_3	S_1	S_2	S_3	S_1	S_2	S_3
Polska	1	duplicated logical network	3	36	no	n/a	736	787	n/a	347	374	n/a	389	413
	2		3	36	yes	534	519	527	225	176	189	309	343	338
	3		4	48	no	n/a	570	585	n/a	273	285	n/a	297	300
	4		4	48	yes	479	443	461	225	176	189	254	267	272
	5		5	60	no	489	465	479	245	224	235	244	241	244
	6		5	60	yes	447	413	424	225	176	189	222	237	235
	7	partitioned logical network	4	24	no	n/a	759	761	n/a	353	392	n/a	406	369
	8		4	24	yes	617	571	615	258	223	251	359	348	364
	9		5	30	no	562	569	580	273	271	277	289	298	303
	10		5	30	yes	516	496	499	232	194	219	284	302	280
	11		6	36	no	537	493	510	272	224	237	265	269	273
	12		6	36	yes	485	431	466	233	181	199	252	250	267
	13		7	42	no	476	451	482	241	212	233	235	239	249
	14		7	42	yes	467	421	434	228	182	198	239	239	236
NSF	1	duplicated logical network	3	42	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	2		3	42	yes	793	735	793	305	285	331	488	450	462
	3		4	56	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	4		4	56	yes	687	659	712	305	285	331	382	374	381
	5		5	70	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	6		5	70	yes	649	639	698	305	285	331	344	354	367
	7	partitioned logical network	4	28	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	8		4	28	yes	981	834	1079	396	341	455	585	494	623
	9		5	35	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	10		5	35	yes	757	733	833	335	303	364	422	430	468
	11		6	42	no	n/a	781	n/a	n/a	359	n/a	n/a	422	n/a
	12		6	42	yes	709	672	757	319	287	351	390	384	406
	13		7	49	no	791	772	n/a	405	351	n/a	386	421	n/a
	14		7	49	yes	677	641	716	316	286	341	361	354	376
Spain	1	duplicated logical network	3	42	no	n/a	1189	n/a	n/a	546	n/a	n/a	643	n/a
	2		3	42	yes	713	795	755	265	254	317	448	541	438
	3		4	56	no	n/a	852	n/a	n/a	409	n/a	n/a	443	n/a
	4		4	56	yes	673	671	693	265	254	317	408	417	375
	5		5	70	no	782	757	785	347	355	389	435	402	396
	6		5	70	yes	776	611	664	289	254	317	486	357	347
	7	partitioned logical network	4	28	no	n/a	1173	n/a	n/a	594	n/a	n/a	580	n/a
	8		4	28	yes	988	908	940	359	326	393	629	581	547
	9		5	35	no	1033	947	923	441	438	441	592	509	482
	10		5	35	yes	776	728	780	292	288	343	484	440	437
	11		6	42	no	764	821	783	340	384	374	424	436	409
	12		6	42	yes	714	686	730	278	273	331	436	413	399
	13		7	49	no	735	732	729	354	346	365	381	386	364
	14		7	49	yes	653	638	700	273	264	325	379	373	375
Italia14	1	duplicated logical network	3	42	no	998	952	997	468	441	480	530	511	517
	2		3	42	yes	603	564	629	215	201	223	388	363	406
	3		4	56	no	738	720	753	365	348	375	373	372	378
	4		4	56	yes	577	556	566	215	201	223	362	355	343
	5		5	70	no	645	625	650	311	297	319	334	328	331
	6		5	70	yes	535	514	543	215	201	223	320	313	320
	7	partitioned logical network	4	28	no	942	859	874	487	445	483	455	414	391
	8		4	28	yes	781	783	777	301	275	275	480	508	502
	9		5	35	no	794	772	792	386	377	388	407	395	405
	10		5	35	yes	644	613	655	232	209	241	412	405	414
	11		6	42	no	699	684	700	338	325	339	361	358	361
	12		6	42	yes	587	552	607	230	209	238	357	343	368
	13		7	49	no	603	587	614	287	285	296	316	302	317
	14		7	49	yes	540	526	561	227	204	233	313	323	328
EPAN	1	duplicated logical network	3	48	no	2270	1951	n/a	992	928	n/a	1278	1023	n/a
	2		3	48	yes	1225	1209	1173	422	396	539	803	812	634
	3		4	64	no	1473	1385	n/a	713	628	n/a	760	757	n/a
	4		4	64	yes	1007	994	1102	422	396	539	585	598	563
	5		5	80	no	1268	1224	n/a	636	615	n/a	632	609	n/a
	6		5	80	yes	991	967	1052	422	396	539	569	571	513
	7	partitioned logical network	4	32	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	8		4	32	yes	1391	1399	1500	527	582	676	864	817	824
	9		5	40	no	1484	1495	n/a	686	736	n/a	798	759	n/a
	10		5	40	yes	1218	1189	1213	456	460	560	763	729	653
	11		6	48	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	12		6	48	yes	1069	1075	1123	453	415	555	617	660	569
	13		7	56	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	14		7	56	yes	1045	988	1087	436	418	545	609	570	542

Table 6.10: Total resources used for QoR class-1 in the two protection configurations (Bottom Layer results shown in Table 6.3.

Network	Scenario	Protection Configuration						% Percentage of the difference		
		Bottom Layer			Top Layer					
		S ₁	S ₂	S ₃	S ₁	S ₂	S ₃	S ₁	S ₂	S ₃
Polska	1	844	733	772	n/a	736	787	n/a	-0.4	-1.9
	2	685	581	586	534	519	527	22.0	10.7	10.1
	3	672	595	616	n/a	570	585	n/a	4.2	5.0
	4	601	518	533	479	443	461	20.3	14.5	13.5
	5	567	495	518	489	465	479	13.8	6.1	7.5
	6	547	481	488	447	413	424	18.3	14.1	13.1
	7	946	831	877	n/a	759	761	n/a	8.7	13.2
	8	811	748	745	617	571	615	23.9	23.7	17.4
	9	704	637	664	562	569	580	20.2	10.7	12.7
	10	685	627	613	516	496	499	24.7	20.9	18.6
	11	623	564	588	537	493	510	13.8	12.6	13.3
	12	610	548	572	485	431	466	20.5	21.4	18.5
	13	580	520	536	476	451	482	17.9	13.3	10.1
	14	572	514	518	467	421	434	18.4	18.1	16.2
	Avg.	675	599	616	510	524	544	19.4	12.7	12.0
NSF	1	1324	1297	1350	n/a	n/a	n/a	n/a	n/a	n/a
	2	1171	1110	1199	793	735	793	32.3	33.8	33.9
	3	1072	1026	1099	n/a	n/a	n/a	n/a	n/a	n/a
	4	993	962	1029	687	659	712	30.8	31.5	30.8
	5	946	905	971	n/a	n/a	n/a	n/a	n/a	n/a
	6	913	876	930	649	639	698	29.0	27.1	24.9
	7	1613	1564	1600	n/a	n/a	n/a	n/a	n/a	n/a
	8	1472	1458	1520	981	834	1079	33.4	42.8	29.0
	9	1230	1189	1241	n/a	n/a	n/a	n/a	n/a	n/a
	10	1144	1105	1164	757	733	833	33.8	33.7	28.5
	11	1037	1002	1070	n/a	781	n/a	n/a	22.1	n/a
	12	1042	997	1046	709	672	757	32.0	32.6	27.6
	13	977	937	1008	791	772	n/a	19.0	17.6	n/a
	14	946	917	971	677	641	716	28.4	30.1	26.3
	Avg.	1134	1096	1157	756	718	798	29.8	30.1	28.7
Spain	1	1219	1173	1325	n/a	1189	n/a	n/a	-1.3	n/a
	2	890	887	975	713	795	755	19.8	10.4	22.6
	3	919	888	972	n/a	852	n/a	n/a	4.1	n/a
	4	815	806	877	673	671	693	17.4	16.8	21.0
	5	839	797	879	782	757	785	6.8	5.0	10.7
	6	768	744	810	776	611	664	-0.9	17.9	18.1
	7	1343	1287	1393	n/a	1173	n/a	n/a	8.9	n/a
	8	1125	1076	1251	988	908	940	12.2	15.7	24.8
	9	1066	986	1080	1033	947	923	3.1	4.0	14.5
	10	952	968	1030	776	728	780	18.5	24.8	24.3
	11	919	887	943	764	821	783	16.9	7.5	17.0
	12	903	857	918	714	686	730	20.9	20.0	20.5
	13	828	813	866	735	732	729	11.3	10.0	15.9
	14	814	796	853	653	638	700	19.8	19.9	17.9
	Avg.	957	926	1012	782	822	771	13.3	11.7	18.8
Italia14	1	988	1017	1006	998	952	997	-1.0	6.4	1.0
	2	620	582	654	603	564	629	2.7	3.1	3.8
	3	777	734	773	738	720	753	5.0	1.9	2.5
	4	595	571	620	577	556	566	3.1	2.7	8.6
	5	671	642	681	645	625	650	3.8	2.6	4.5
	6	577	550	593	535	514	543	7.1	6.6	8.4
	7	1101	1052	1093	942	859	874	14.5	18.3	20.0
	8	890	709	817	781	783	777	12.3	-10.5	4.9
	9	853	826	853	794	772	792	6.9	6.5	7.1
	10	706	671	735	644	613	655	8.8	8.7	10.9
	11	732	719	741	699	684	700	4.5	4.9	5.5
	12	666	621	675	587	552	607	11.8	11.0	10.1
	13	656	636	663	603	587	614	8.0	7.7	7.5
	14	620	592	636	540	526	561	13.0	11.1	11.9
	Avg.	747	709	753	692	665	694	7.2	5.8	7.6
EPAN	1	1994	1917	2140	2270	1951	n/a	-13.8	-1.7	n/a
	2	1508	1431	1566	1225	1209	1173	18.8	15.5	25.1
	3	1541	1481	1699	1473	1385	n/a	4.5	6.5	n/a
	4	1323	1272	1372	1007	994	1102	23.9	21.9	19.6
	5	1395	1302	1499	1268	1224	n/a	9.1	6.0	n/a
	6	1219	1169	1284	991	967	1052	18.7	17.3	18.1
	7	2209	2163	2396	n/a	n/a	n/a	n/a	n/a	n/a
	8	2045	1865	1904	1391	1399	1500	32.0	25.0	21.2
	9	1757	1635	1837	1484	1495	n/a	15.5	8.6	n/a
	10	1604	1526	1554	1218	1189	1213	24.0	22.1	21.9
	11	1487	1426	1562	n/a	n/a	n/a	n/a	n/a	n/a
	12	1412	1404	1422	1069	1075	1123	24.3	23.4	21.0
	13	1329	1283	1399	n/a	n/a	n/a	n/a	n/a	n/a
	14	1334	1269	1344	1045	988	1087	21.6	22.1	19.2
	Avg.	1583	1510	1641	1313	1261	1179	16.2	15.1	20.9

better resource utilization. Table 6.10 compares the total resources required for QoR class-1 in the two protection configurations. Take the Polska network as an example, we found that providing a path-pair for each flow (QoR class-1) at the upper layer requires, on average, 19.4, 12.7, and 12.0% less resources than the case with path-pairs given at the physical layer for S_1 , S_2 , and S_3 respectively. However, we see no clear pattern across all cases of whether the WP or the BP contributes more to these differences. One also can see in the table that when the logical topology is preselected the difference in the total resources becomes larger. In addition, the denser the physical network (*e.g.*, Italia14), the lower the difference between the two configurations.

Routing on Logical Subnetworks: For routing lower classes flows, we solve the LP given by (6.47)-(6.51) for the 14 scenarios for spine S_2 . We compute the resources required by single path on X (*i.e.*, class-3) and single path on Y (*i.e.*, class-4) with demand $d_\phi^{mn} = 1$, $\forall mn, \phi = 3, 4$. Table 6.11 shows the resources used by both QoR class-3 and class-4 traffic for the five test networks. As mentioned earlier that these paths are not required to be disjoint, and hence might show a lower sum than QoR class-1. In addition, Table 6.12 compares the resources for single path on Y (class-4) obtained here for S_2 to results obtained for class-4 in Table 6.3. The negative sign indicates less resources are needed for the protection configuration in the top layer, which only appears in the partitioned scenarios. One can see that the resources needed for providing QoR class-4 is not determined by the protection configuration and is very dependent on the logical topology and the problem type.

Availability: After solving the optimization problems, we use path and mapping information to calculate the logical links and flows expected downtimes. We only consider link availability (*i.e.*, assuming nodes are perfect). As noted earlier we use $a_e=0.999$ for links on the spine ($e \in \mathcal{E}_S$) and $a_e=0.99$ for links off the spine ($e \in \mathcal{E}_P$ but $e \notin \mathcal{E}_S$). Table 6.13 shows the average and maximum expected logical link downtime for each subnetwork, scenario, spine, and network. First, the same observation as made about logical graph density and randomness versus resource efficiency holds true with respect to downtime values. Second, it is clear from Table 6.13 that subnetwork X , which is mapped to the spine, achieves much smaller downtime values compared to subnetwork Y . Third, using these two subnetworks and a single protection scheme, we were able to create three different QoR classes of avail-

Table 6.11: Resource use for unconstrained routing on logical subnetworks X and Y for spine S_2 . (* path-pair is not disjoint)

Network	Scenario	Total lengths (full-mesh) of minimum weight paths on X and Y on S_2		
		Paths on X QoR class-3	Paths on Y QoR class-4	total *
Polska	1	335	361	696
	2	176	298	474
	3	269	260	529
	4	176	213	389
	5	222	208	430
	6	176	196	372
	7	348	351	699
	8	223	307	530
	9	270	271	541
	10	194	278	472
	11	223	240	463
	12	180	229	409
	13	212	224	436
	14	182	219	401
NSF	1	n/a	n/a	n/a
	2	285	385	670
	3	n/a	n/a	n/a
	4	285	323	608
	5	n/a	n/a	n/a
	6	285	258	543
	7	n/a	n/a	n/a
	8	285	351	636
	9	n/a	n/a	n/a
	10	285	302	587
	11	343	340	683
	12	286	301	587
	13	333	329	662
	14	285	276	561
Spain	1	532	611	1143
	2	254	455	709
	3	400	405	806
	4	254	328	582
	5	351	362	713
	6	254	276	530
	7	432	373	805
	8	268	362	630
	9	389	384	773
	10	254	325	579
	11	347	328	675
	12	260	315	575
	13	325	289	613
	14	254	278	532
Italia14	1	438	497	935
	2	201	348	549
	3	345	346	691
	4	201	324	525
	5	296	311	606
	6	201	279	480
	7	377	316	693
	8	201	350	551
	9	335	346	681
	10	201	332	533
	11	305	313	618
	12	201	280	481
	13	268	262	529
	14	201	271	472
EPAN	1	851	918	1769
	2	396	654	1050
	3	623	713	1336
	4	396	515	911
	5	601	542	1143
	6	396	465	861
	7	n/a	n/a	n/a
	8	401	436	837
	9	630	594	1224
	10	400	491	891
	11	n/a	n/a	n/a
	12	396	468	864
	13	n/a	n/a	n/a
	14	398	435	833

Table 6.12: A comparison of the resource use for single unprotected paths for spine S_2 between the two configurations.

Network	Scenario	Protection Configuration		% of the difference
		Dedicated logical network (<i>bottom</i>) QoS class-4	Routing on logical subnetwork Y (<i>top</i>) QoS class-4	
Polska	1	275	361	23.8
	2	168	298	43.6
	3	218	260	16.2
	4	160	213	24.9
	5	180	208	13.5
	6	154	196	21.4
	7	400	351	-14.0
	8	316	307	-2.9
	9	292	271	-7.7
	10	253	278	9.0
	11	248	240	-3.3
	12	199	229	13.1
	13	221	224	1.3
	14	196	219	10.5
NSF	1	419	n/a	n/a
	2	250	385	35.1
	3	340	n/a	n/a
	4	235	323	27.1
	5	291	n/a	n/a
	6	232	258	10.1
	7	544	n/a	n/a
	8	409	351	-16.5
	9	423	n/a	n/a
	10	364	302	-20.6
	11	379	340	-11.5
	12	319	301	-5.8
	13	350	329	-6.3
	14	288	276	-4.4
Spain	1	439	611	28.1
	2	242	455	46.9
	3	318	405	21.5
	4	237	328	27.8
	5	293	362	19.2
	6	226	276	18.1
	7	587	373	-57.6
	8	523	362	-44.5
	9	497	384	-29.5
	10	353	325	-8.6
	11	412	328	-25.7
	12	336	315	-6.9
	13	342	289	-18.4
	14	291	278	-4.8
Italia14	1	391	497	21.3
	2	196	348	43.6
	3	277	346	19.8
	4	196	324	39.5
	5	244	311	21.6
	6	190	279	32.1
	7	546	395	-38.2
	8	527	376	-40.0
	9	410	346	-18.8
	10	349	332	-5.1
	11	348	313	-11.0
	12	307	280	-9.7
	13	294	262	-12.4
	14	284	271	-4.6
EPAN	1	726	918	20.9
	2	376	654	42.5
	3	542	713	24.1
	4	358	515	30.6
	5	490	542	9.6
	6	352	465	24.1
	7	1032	n/a	n/a
	8	804	436	-84.5
	9	817	594	-37.6
	10	615	491	-25.3
	11	630	n/a	n/a
	12	485	468	-3.5
	13	581	n/a	n/a
	14	446	435	-2.7

Table 6.13: Average and maximum logical link expected downtime.

Network	Scenario	Problem type	regular graph degree k	total no. of logical links	preselected logical links	Average/maximum expected logical link Downtime (hours/year)					
						Subnetwork- X			Subnetwork- Y		
						S1	S2	S3	S1	S2	S3
Polska	1	duplicated logical network	3	36	no	n/a	21 / 38	23 / 42	n/a	148 / 356	136 / 314
	2		3	36	yes	9 / 13	9 / 14	9 / 13	57 / 191	106 / 325	102 / 282
	3		4	48	no	n/a	21 / 39	22 / 45	n/a	127 / 303	117 / 260
	4		4	48	yes	10 / 26	10 / 23	9 / 15	67 / 195	100 / 325	100 / 270
	5		5	60	no	16 / 44	19 / 38	19 / 39	73 / 191	110 / 316	103 / 260
	6		5	60	yes	9 / 11	9 / 10	9 / 14	68 / 198	102 / 311	96 / 270
	7	partitioned logical network	4	24	no	n/a / n/a	21 / 38	21 / 44	n/a / n/a	149 / 350	117 / 269
	8		4	24	yes	14 / 45	13 / 36	14 / 37	94 / 197	134 / 339	123 / 345
	9		5	30	no	15 / 44	20 / 36	18 / 32	96 / 191	123 / 315	118 / 271
	10		5	30	yes	15 / 49	13 / 36	13 / 34	89 / 196	133 / 337	116 / 284
	11		6	36	no	20 / 46	18 / 36	18 / 39	89 / 215	116 / 292	114 / 269
	12		6	36	yes	17 / 49	11 / 29	13 / 32	83 / 173	119 / 303	107 / 247
	13		7	42	no	18 / 35	15 / 35	19 / 39	83 / 183	110 / 309	107 / 269
	14		7	42	yes	17 / 51	12 / 29	13 / 34	88 / 196	112 / 284	104 / 260
NSF	1	duplicated logical network	3	42	no	n/a	n/a	n/a	n/a	n/a	n/a
	2		3	42	yes	10 / 26	10 / 24	10 / 28	69 / 270	74 / 267	63 / 260
	3		4	56	no	n/a	n/a	n/a	n/a	n/a	n/a
	4		4	56	yes	9 / 19	9 / 14	10 / 24	72 / 236	73 / 253	69 / 250
	5		5	70	no	n/a	n/a	n/a	n/a	n/a	n/a
	6		5	70	yes	10 / 23	10 / 19	13 / 37	69 / 222	77 / 299	82 / 250
	7	partitioned logical network	4	28	no	n/a	n/a	n/a	n/a	n/a	n/a
	8		4	28	yes	15 / 50	13 / 35	14 / 52	110 / 311	114 / 271	97 / 247
	9		5	35	no	n/a	n/a	n/a	n/a	n/a	n/a
	10		5	35	yes	14 / 44	13 / 33	14 / 51	97 / 257	107 / 301	96 / 253
	11		6	42	no	n/a	18 / 35	n/a	n/a	104 / 269	n/a
	12		6	42	yes	14 / 42	13 / 38	15 / 44	104 / 238	109 / 301	101 / 269
	13		7	49	no	21 / 44	20 / 44	n/a	106 / 191	131 / 345	n/a
	14		7	49	yes	13 / 37	13 / 33	15 / 35	103 / 240	99 / 316	102 / 241
Spain	1	duplicated logical network	3	42	no	n/a	24 / 42	n/a	n/a	147 / 355	n/a
	2		3	42	yes	10 / 33	10 / 24	11 / 44	75 / 265	101 / 321	82 / 250
	3		4	56	no	n/a	22 / 40	n/a	n/a	127 / 330	n/a
	4		4	56	yes	11 / 23	11 / 31	12 / 33	85 / 275	95 / 321	87 / 256
	5		5	70	no	20 / 52	20 / 40	20 / 52	92 / 209	120 / 332	118 / 286
	6		5	70	yes	12 / 35	12 / 33	10 / 33	107 / 262	94 / 298	89 / 270
	7	partitioned logical network	4	28	no	n/a	22 / 39	n/a	n/a	128 / 359	n/a
	8		4	28	yes	14 / 38	14 / 38	15 / 45	104 / 298	130 / 367	108 / 260
	9		5	35	no	21 / 52	19 / 35	20 / 52	107 / 209	133 / 306	122 / 269
	10		5	35	yes	12 / 35	12 / 33	13 / 37	104 / 245	118 / 318	115 / 252
	11		6	42	no	16 / 39	21 / 42	18 / 44	107 / 239	127 / 323	111 / 269
	12		6	42	yes	12 / 33	12 / 35	12 / 35	114 / 250	125 / 341	121 / 275
	13		7	49	no	19 / 41	20 / 44	20 / 48	101 / 280	111 / 355	100 / 273
	14		7	49	yes	12 / 35	13 / 33	14 / 42	103 / 263	118 / 324	124 / 279
italia14	1	duplicated logical network	3	42	no	20 / 30	19 / 26	21 / 33	157 / 324	161 / 316	150 / 272
	2		3	42	yes	10 / 21	9 / 18	9 / 16	123 / 253	111 / 272	112 / 272
	3		4	56	no	20 / 35	19 / 26	20 / 35	135 / 286	136 / 316	132 / 277
	4		4	56	yes	9 / 21	9 / 9	9 / 12	106 / 240	124 / 287	105 / 272
	5		5	70	no	18 / 33	18 / 26	18 / 35	129 / 292	130 / 301	128 / 274
	6		5	70	yes	10 / 23	10 / 18	10 / 18	110 / 251	121 / 284	109 / 269
	7	partitioned logical network	4	28	no	20 / 35	18 / 26	19 / 35	125 / 307	100 / 260	102 / 191
	8		4	28	yes	13 / 31	12 / 24	12 / 26	118 / 269	144 / 328	126 / 304
	9		5	35	no	18 / 30	17 / 26	18 / 30	129 / 274	138 / 282	129 / 286
	10		5	35	yes	11 / 30	9 / 18	11 / 30	132 / 287	141 / 286	132 / 270
	11		6	42	no	17 / 28	17 / 26	18 / 30	125 / 250	141 / 284	127 / 267
	12		6	42	yes	10 / 26	10 / 16	10 / 26	129 / 284	133 / 284	131 / 308
	13		7	49	no	16 / 31	16 / 26	17 / 35	117 / 275	120 / 301	119 / 260
	14		7	49	yes	11 / 28	9 / 18	10 / 23	116 / 270	135 / 282	120 / 265
EPAN	1	duplicated logical network	3	48	no	30 / 57	28 / 61	n/a	57 / 412	156 / 362	n/a
	2		3	48	yes	12 / 37	10 / 33	9 / 14	86 / 316	93 / 347	60 / 270
	3		4	64	no	27 / 61	24 / 39	n/a	61 / 290	144 / 262	n/a
	4		4	64	yes	11 / 30	10 / 28	11 / 54	30 / 286	96 / 340	76 / 257
	5		5	80	no	25 / 61	24 / 48	n/a	61 / 290	127 / 324	n/a
	6		5	80	yes	11 / 37	11 / 38	15 / 58	37 / 281	103 / 360	77 / 269
	7	partitioned logical network	4	32	no	n/a	n/a	n/a	n/a	n/a	n/a
	8		4	32	yes	16 / 48	17 / 50	19 / 61	48 / 281	115 / 345	108 / 279
	9		5	40	no	n/a	n/a	n/a	n/a	n/a	n/a
	10		5	40	yes	13 / 38	14 / 42	15 / 51	38 / 281	137 / 364	115 / 286
	11		6	48	no	n/a	n/a	n/a	n/a	n/a	n/a
	12		6	48	yes	14 / 44	14 / 47	19 / 66	44 / 270	136 / 345	118 / 274
	13		7	56	no	n/a	n/a	n/a	n/a	n/a	n/a
	14		7	56	yes	13 / 37	14 / 38	17 / 59	37 / 289	127 / 325	115 / 270

Table 6.14: Average and maximum logical flows expected downtime.

Network	Scenario	Problem type	regular graph degree k	total no. of logical links	preselected logical links	Average/maximum expected Flow Downtime (hours/year)								
						Physically disjoint path-pair QoR class-1			single path on subnet-X QoR class-3			single path on subnet-Y QoR class-4		
						S1	S2	S3	S1	S2	S3	S1	S2	S3
Polska	1	duplicated logical network	3	36	no	n/a	2.13 / 6.72	2.16 / 6.62	n/a	46 / 97	49 / 107	n/a	350 / 759	334 / 715
			3	36	yes	0.61 / 2.23	0.75 / 2.25	0.77 / 2.63	30 / 70	23 / 44	25 / 52	172 / 379	249 / 555	244 / 515
			4	48	no	n/a	1.19 / 3.76	1.09 / 3.06	n/a	36 / 67	38 / 75	n/a / n/a	261 / 552	237 / 476
			4	48	yes	0.56 / 2.12	0.66 / 2.10	0.62 / 2.06	30 / 70	23 / 44	25 / 52	157 / 318	222 / 542	200 / 450
			5	60	no	0.62 / 1.86	0.77 / 2.16	0.72 / 2.09	31 / 70	30 / 56	31 / 61	158 / 361	210 / 412	192 / 394
			5	60	yes	0.52 / 1.83	0.58 / 1.87	0.57 / 1.77	30 / 70	23 / 44	25 / 52	147 / 307	200 / 444	186 / 391
	2	partitioned logical network	4	24	no	n/a	2.04 / 7.14	1.79 / 7.23	n/a	47 / 105	52 / 150	n/a	378 / 898	314 / 813
			4	24	yes	0.94 / 3.55	1.12 / 4.51	1.14 / 4.46	34 / 91	30 / 75	33 / 81	237 / 516	308 / 744	273 / 659
			5	30	no	n/a / n/a	1.14 / 4.15	1.06 / 3.31	n/a / n/a	36 / 72	37 / 75	n/a / n/a	273 / 619	250 / 547
			5	30	yes	0.68 / 2.31	0.82 / 2.84	0.79 / 2.58	31 / 71	26 / 59	29 / 57	194 / 408	268 / 567	227 / 490
			6	36	no	0.74 / 2.33	0.83 / 2.44	0.77 / 2.29	36 / 72	30 / 56	31 / 60	177 / 349	240 / 535	214 / 470
			6	36	yes	0.60 / 2.18	0.63 / 1.96	0.64 / 1.84	31 / 71	24 / 49	26 / 52	164 / 322	220 / 440	205 / 399
			7	42	no	0.59 / 1.40	0.71 / 1.82	0.73 / 2.11	31 / 70	28 / 51	31 / 56	159 / 291	214 / 407	205 / 430
			7	42	yes	0.54 / 1.71	0.59 / 1.74	0.60 / 1.78	30 / 70	24 / 44	26 / 52	157 / 306	206 / 391	192 / 382
NSF	1	duplicated logical network	3	42	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			3	42	yes	0.82 / 3.73	0.78 / 2.67	0.72 / 2.83	29 / 61	27 / 54	32 / 79	228 / 629	227 / 520	197 / 476
			4	56	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			4	56	yes	0.65 / 2.21	0.70 / 2.43	0.67 / 2.69	29 / 61	27 / 52	32 / 79	187 / 395	206 / 474	182 / 472
			5	70	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			5	70	yes	0.57 / 2.02	0.69 / 2.52	0.63 / 2.51	29 / 61	27 / 52	32 / 79	168 / 377	199 / 476	174 / 442
	2	partitioned logical network	4	28	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			4	28	yes	1.18 / 5.88	1.17 / 4.70	1.70 / 6.31	38 / 102	32 / 87	44 / 122	254 / 687	299 / 785	329 / 920
			5	35	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			5	35	yes	0.81 / 3.06	0.86 / 3.08	0.95 / 3.68	32 / 72	29 / 65	35 / 80	217 / 489	244 / 559	234 / 595
			6	42	no	n/a	0.99 / 3.64	n/a	n/a	34 / 79	n/a	n/a	251 / 603	n/a
			6	42	yes	0.73 / 2.53	0.74 / 2.64	0.78 / 3.32	31 / 68	28 / 54	34 / 79	210 / 459	229 / 522	205 / 522
			7	49	no	0.94 / 2.53	0.99 / 3.00	n/a	39 / 70	34 / 61	n/a	210 / 446	254 / 521	n/a
			7	49	yes	0.67 / 2.53	0.66 / 2.31	0.68 / 2.35	30 / 61	28 / 52	33 / 79	192 / 411	200 / 439	186 / 429
Spain	1	duplicated logical network	3	42	no	n/a	2.62 / 8.17	n/a	n/a	52 / 103	n/a	n/a	379 / 863	n/a
			3	42	yes	0.65 / 2.03	0.78 / 2.68	0.85 / 3.22	25 / 52	24 / 44	30 / 61	211 / 416	255 / 606	230 / 591
			4	56	no	n/a	1.30 / 4.39	n/a	n/a	39 / 75	n/a	n/a	268 / 625	n/a
			4	56	yes	0.59 / 1.89	0.66 / 2.27	0.72 / 2.41	25 / 52	24 / 44	31 / 61	192 / 380	213 / 485	198 / 446
			5	70	no	0.82 / 2.53	0.95 / 2.86	1.04 / 3.05	33 / 61	34 / 61	37 / 79	201 / 379	225 / 479	226 / 446
			5	70	yes	0.80 / 2.70	0.61 / 1.89	0.66 / 2.45	28 / 63	24 / 44	30 / 61	244 / 478	201 / 458	185 / 406
	2	partitioned logical network	4	28	no	n/a	2.34 / 12.2	n/a	n/a	57 / 133	n/a	n/a	360 / 973	n/a
			4	28	yes	1.29 / 5.31	1.21 / 5.24	1.34 / 4.98	34 / 89	31 / 82	38 / 89	312 / 829	324 / 736	297 / 720
			5	35	no	1.28 / 4.04	1.45 / 4.63	1.34 / 4.05	42 / 113	42 / 85	42 / 105	254 / 578	301 / 618	268 / 611
			5	35	yes	0.82 / 2.96	0.87 / 3.05	0.91 / 2.90	28 / 65	28 / 73	33 / 66	246 / 509	268 / 603	242 / 555
			6	42	no	0.85 / 2.29	1.13 / 3.47	0.95 / 2.96	33 / 70	37 / 77	36 / 70	231 / 421	263 / 600	233 / 495
			6	42	yes	0.72 / 2.30	0.78 / 2.68	0.82 / 2.60	27 / 63	26 / 59	32 / 65	233 / 530	248 / 568	225 / 513
			7	49	no	0.80 / 2.11	0.90 / 2.87	0.82 / 1.91	34 / 67	33 / 65	35 / 65	201 / 424	232 / 555	207 / 375
			7	49	yes	0.62 / 2.14	0.67 / 2.02	0.80 / 2.44	26 / 59	25 / 58	31 / 63	201 / 451	223 / 487	220 / 461
Italia14	1	duplicated logical network	3	42	no	2.19 / 6.94	2.20 / 7.02	2.23 / 7.28	45 / 91	42 / 87	46 / 91	375 / 834	390 / 842	372 / 845
			3	42	yes	0.65 / 1.74	0.50 / 1.17	0.70 / 1.97	21 / 35	19 / 26	21 / 35	255 / 488	211 / 402	264 / 510
			4	56	no	1.19 / 3.60	1.19 / 3.76	1.24 / 4.10	35 / 68	33 / 66	36 / 72	272 / 559	279 / 601	273 / 637
			4	56	yes	0.58 / 1.42	0.53 / 1.40	0.58 / 1.82	21 / 35	19 / 26	21 / 35	232 / 428	222 / 468	223 / 469
			5	70	no	0.88 / 2.44	0.86 / 2.44	0.86 / 2.36	30 / 54	29 / 54	31 / 56	239 / 544	244 / 532	231 / 501
			5	70	yes	0.54 / 1.61	0.51 / 1.39	0.56 / 1.54	21 / 35	19 / 26	21 / 35	217 / 469	214 / 462	216 / 469
	2	partitioned logical network	4	28	no	1.85 / 4.83	1.59 / 4.09	1.57 / 3.92	47 / 105	43 / 96	46 / 87	362 / 765	341 / 677	307 / 611
			4	28	yes	1.12 / 4.85	1.15 / 4.15	1.00 / 3.73	29 / 65	26 / 65	26 / 63	321 / 697	362 / 821	329 / 740
			5	35	no	1.29 / 3.84	1.26 / 3.58	1.21 / 3.45	37 / 75	36 / 66	37 / 80	307 / 635	314 / 633	292 / 588
			5	35	yes	0.75 / 2.90	0.71 / 1.87	0.76 / 2.91	22 / 58	20 / 35	23 / 54	286 / 649	306 / 597	281 / 565
			6	42	no	0.99 / 2.79	1.05 / 2.80	0.97 / 3.07	33 / 66	31 / 59	33 / 65	269 / 542	293 / 580	264 / 512
			6	42	yes	0.62 / 1.93	0.60 / 1.80	0.65 / 2.33	22 / 58	20 / 38	23 / 52	248 / 540	258 / 531	251 / 524
			7	49	no	0.73 / 2.23	0.76 / 2.30	0.76 / 2.30	28 / 54	27 / 49	28 / 56	237 / 471	247 / 516	238 / 486
			7	49	yes	0.54 / 1.38	0.56 / 1.29	0.57 / 1.59	22 / 49	20 / 31	22 / 51	216 / 454	248 / 533	224 / 444
EPAN	1	duplicated logical network	3	48	no	4.56 / 23.9	3.73 / 11.9	n/a	72 / 195	67 / 143	n/a	504 / 1262	441 / 952	n/a
			3	48	yes	1.07 / 4.01	1.10 / 3.40	0.77 / 2.60	34 / 70	29 / 61	34 / 70	256 / 643	309 / 611	197 / 443
			4	64	no	1.82 / 6.51	1.85 / 4.66	n/a	52 / 113	46 / 77	n/a	285 / 677	325 / 517	n/a
			4	64	yes	0.78 / 2.63	0.89 / 2.87	0.94 / 3.91	31 / 70	28 / 59	39 / 96	211 / 478	257 / 580	206 / 513
			5	80	no	1.29 / 4.72	1.55 / 5.27	n/a	46 / 100	45 / 92	n/a	227 / 525	271 / 652	n/a
			5	80	yes	0.76 / 2.73	0.73 / 3.20	0.89 / 3.69	31 / 70	28 / 65	39 / 96	208 / 451	210 / 547	193 / 429
	2	partitioned logical network	4	32	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			4	32	yes	1.36 / 5.14	1.77 / 7.61	1.89 / 8.86	38 / 96	42 / 118	49 / 120	302 / 737	348 / 881	322 / 799
			5	40	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			5	40	yes	1.03 / 4.12	1.25 / 5.08	1.22 / 4.73	33 / 73	34 / 87	41 / 99	268 / 621	309 / 751	260 / 619
			6	48	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			6	48	yes	0.91 / 3.41	0.99 / 3.43	1.06 / 3.90	33 / 70	30 / 66	40 / 96	233 / 516	274 / 616	233 / 522
			7	56	no	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
			7	56	yes	0.81 / 3.02	0.92 / 2.86	0.97 / 3.63	32 / 70	30 / 63	40 / 96	221 / 550	251 / 573	216 / 474

ability as shown in Table 6.14. From Table 6.14 one can see for the Polska network and across the scenarios, the average expected downtime per year for QoR class-1 ranges from 31 to 56 minutes for S_1 , 35 to 128 minutes for S_2 , and 34 to 129 minutes for S_3 . The average downtime per year for QoR class-3 ranges from 30 to 36 hours for S_1 , 23 to 47 hours for S_2 , and 25 to 52 hours for S_3 . Lastly for QoR class-4, the average downtime per year is more than 6.1 days for S_1 , 8.3 days for S_2 , and 7.7 days for S_3 . Note that the lowest downtime values, in the five networks, commonly coincide with scenarios 6 and 14. Also with just a few exceptions, the preselected logical link network scenarios achieve much lower downtimes. This shows the impact of the logical network topology on the design. Further, we compare the downtime results with the ones obtained earlier in Section 6.3.4. Figures 6.7 to 6.11 show the average flows expected downtimes (in log-scale) for the two protection configurations on the test networks. One can see the range of the three QoR classes (1, 3, 4) for the configuration with protection at upper layer compared to the two QoR classes (1, 4) of the second configuration.

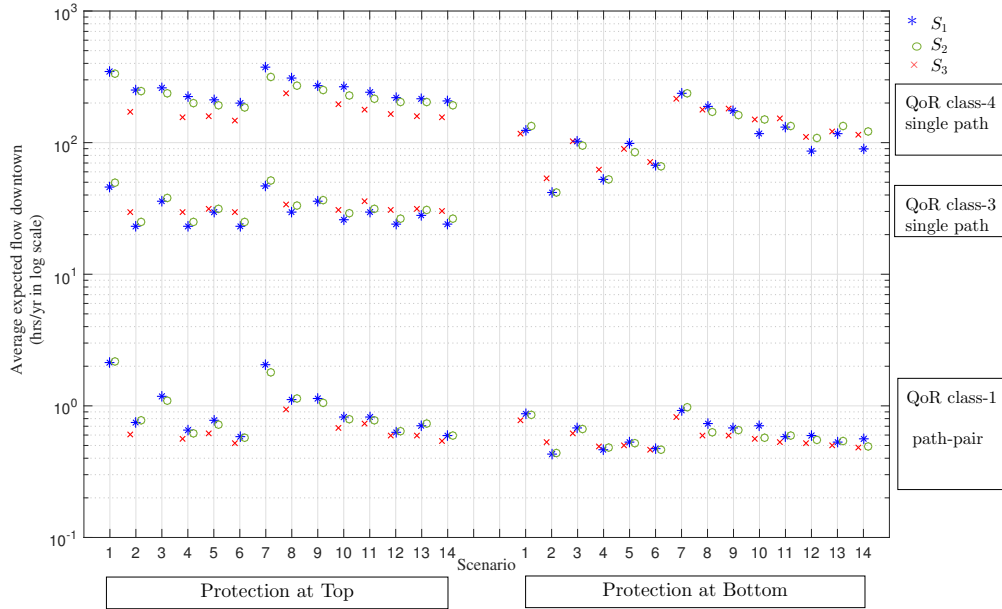


Figure 6.7: Downtime comparison between the two protection configurations on the Polska network.

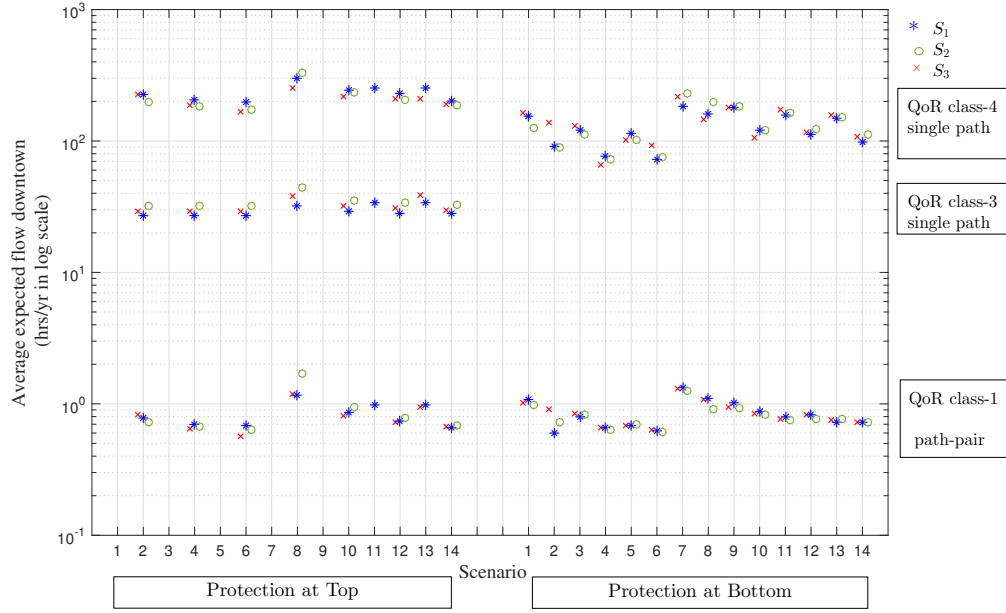


Figure 6.8: Downtime comparison between the two protection configurations on the NSF network.

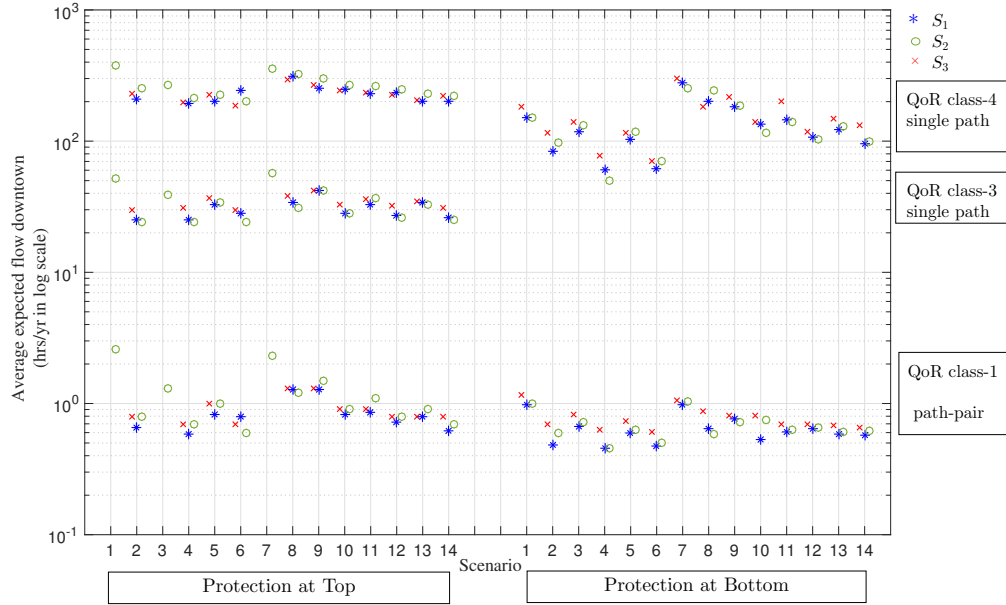


Figure 6.9: Downtime comparison between the two protection configurations on the Spain network.

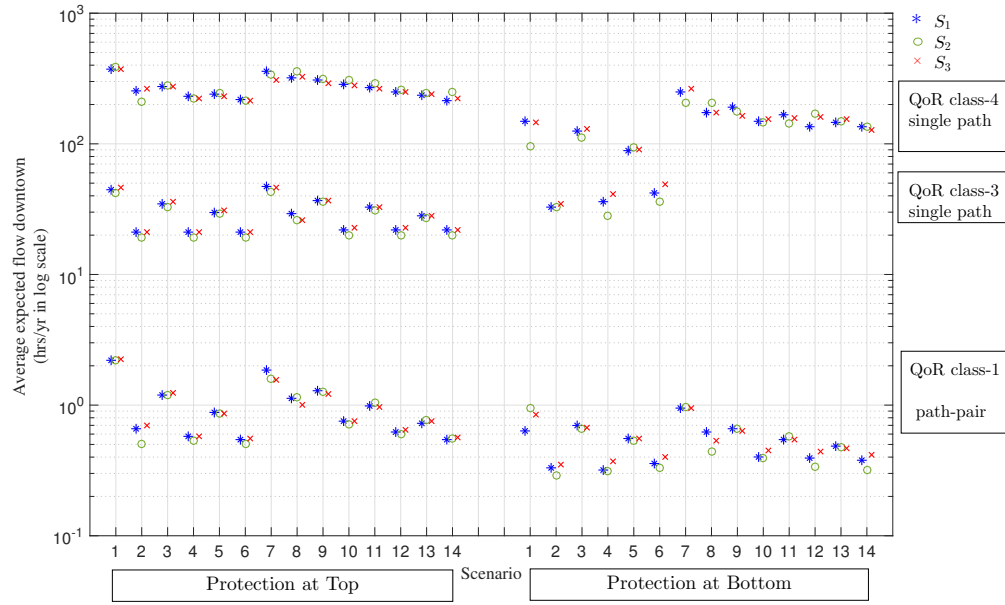


Figure 6.10: Downtime comparison between the two protection configurations on the Italia14 network.

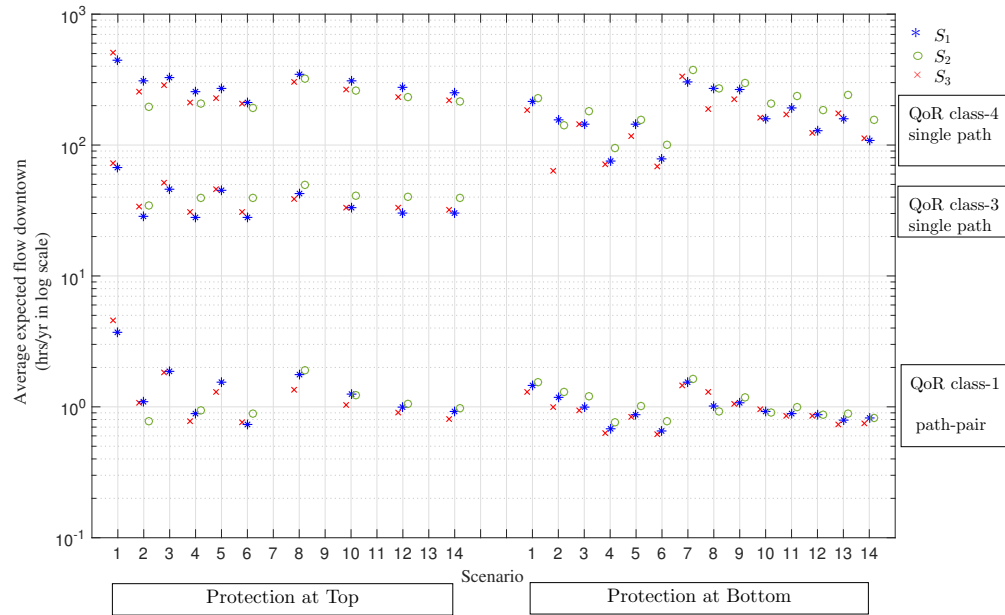


Figure 6.11: Downtime comparison between the two protection configurations on the EPAN network.

The downtime of QoR class-1 and class-4 on the two configurations are roughly in the same range for most scenarios with slightly longer downtime values for the case with protection at the upper layer. Also there is a noticeable impact of the scenario (*i.e.*, logical network topology and problem type) on the downtime values of QoR class-1 for this configuration, while the top layer configuration exhibits a tighter range of downtime values across the different scenarios. The other networks exhibit a similar behavior and there is a great similarity of the downtime values of different networks within each protection configuration.

A comparison with no spine: To get more insight into the resource use of the spine model, we compare our results to a baseline model. The model assumes homogenous link availability at the physical layer in which no spine is embedded and all links have the same availability a_e . In order to have a comparable cost network, a_e is set to a weighted average of the on-spine and off-spine availability, $a_e = \frac{0.999|\mathcal{E}_S| + 0.99|\mathcal{E} - \mathcal{E}_S|}{|\mathcal{E}|}$. In this model, logical links are not constrained to be mapped to a spine, and this would only replace \mathcal{E}_S in our MILP formulation by \mathcal{E}_P . Everything else remains the same. Upper layer flows are routed onto a path-pair, a WP and a BP that are disjoint in the lower layer. The group of all WPs and all BPs are carried over two logical subnetworks and each link is mapped without restriction except the diversity constraint for each path-pair. We solve this model for six of the aforementioned scenarios, and compare them to the results of the spine model in Table 6.9. First, Table 6.15 shows the total resource use for the no-spine model and the percentage of increase in these resources required by the spine model. The percentage of increase for the Polska network is at most 17% for QoR class-1 traffic based on the spine. Also one can see that the second spine S_2 requires only few extra resources more than the baseline model for all the five networks. This confirms that finding the right spine can improve the design resource efficiency and have significant improvement in the downtime values. In practice, only a small fraction (*e.g.*, $\leq 5\%$) of the total traffic is expected to be QoR class-1, thus the potential capacity increase of the spine would be minimal. Lastly, note that in comparison to the no spine model results in Table 6.6, the spine model with protection given at lower layer performs more efficiently than the model of upper layer protection in terms of both resources and downtime values. Second, in terms of availability levels offered by the network, Table 6.16 shows the average and maximum expected downtimes for both class-1, class-3 and class-4

Table 6.15: A comparison of the resource use between the spine and the baseline model.

Network	Scenario	regular graph degree k	total no. of logical links	preselected logical links	Resources required for link-disjoint path-pairs with No Spine	% Percentage of extra resource for link-disjoint path-pairs on the spine		
						S_1	S_2	S_3
Polska	1	3	36	no	672	n/a	9.5	17
	2	3	36	yes	471/491/492 *	13.4	5.9	7.1
	3	4	48	no	531	n/a	7.3	10.2
	4	4	48	yes	448/421/431 *	6.9	5.2	7.0
	5	5	60	no	435	12.4	6.9	10.1
	6	5	60	yes	383/394/396 *	16.7	4.8	7.1
NSF	2	3	42	yes	741/713/737 *	7.1	3.0	7.5
	4	4	56	yes	648/637/645 *	5.9	3.5	10.3
	6	5	70	yes	606/616/632 *	7.1	3.7	10.4
Spain	1	3	42	no	1074	n/a	10.7	n/a
	2	3	42	yes	671/742/696 *	6.2	7.1	8.5
	3	4	56	no	748	n/a	13.9	n/a
	4	4	56	yes	631/643/632 *	6.8	4.4	9.6
	5	5	70	no	709	10.3	6.8	10.7
	6	5	70	yes	609/599/601 *	27.4	2.0	10.3
Italia14	1	3	42	no	900	10.9	5.8	10.8
	2	3	42	yes	580/550/594	4.0	2.7	6.0
	3	4	56	no	664	11.1	8.5	13.5
	4	4	56	yes	538/540/538	7.2	3.0	5.3
	5	5	70	no	596	8.3	4.9	9.1
	6	5	70	yes	510/514/543	5.0	3.6	7.4
EPAN	1	3	48	no	1746	30.0	11.7	n/a
	2	3	48	yes	1142/1128/1058	7.2	7.1	10.8
	3	4	64	no	1298	13.4	6.7	n/a
	4	4	64	yes	949/963/896 *	6.1	3.3	23.0
	5	5	80	no	1149	10.4	6.5	n/a
	6	5	80	yes	922/927/921 *	7.4	4.3	14.2

logical flows in the no-spine model. By comparing these results to the results of the spine model in Table 6.14, we observe that the expected downtimes for class-1 of the no-spine model are on average about 3 times larger than the corresponding results of the spine model for the Polska, Spain and EPAN networks, 2 times for the NSF, and 4 times for Italia14. Class-3 downtimes in the no spine model are roughly 5 times larger than the spine model for the Polska and Spain networks, 4 times larger in the NSF and EPAN networks, and 6.5 times larger in the Italia14 network. Class-4 downtimes are lower with the baseline model than with the spine model in all networks except Italia14. The no spine model, however, achieves lower maximum flow downtimes with respect to class-4. The result of this comparison is identical to the one obtained for Table 6.7 in Section 6.3.4.2.

Table 6.16: The expected downtime of the baseline model (*i.e.*, no spine model).

Network	Scenario	regular graph degree k	total no. of logical links	preselected logical links	Average/maximum expected Flow Downtime (hours/year)								
					Physically disjoint path-pair QoR class-1			single path on subnet-X QoR class-3			single path on subnet-Y QoR class-4		
					S1	S2	S3	S1	S2	S3	S1	S2	S3
Pol ska	1	3	36	no	5.05 / 13.79	5.05 / 13.79	5.05 / 13.79	200 / 381	200 / 381	200 / 381	198 / 370	198 / 370	198 / 370
	2	3	36	yes	2.25 / 5.16	2.51 / 7.67	2.59 / 8.26	145 / 283	145 / 300	153 / 310	138 / 250	147 / 343	141 / 305
	3	4	48	no	3.02 / 7.38	3.02 / 7.38	3.02 / 7.38	158 / 300	158 / 300	158 / 300	156 / 278	156 / 278	156 / 278
	4	4	48	yes	1.83 / 4.53	1.83 / 4.50	1.95 / 4.77	128 / 245	122 / 228	129 / 234	124 / 217	129 / 234	129 / 234
	5	5	60	no	1.98 / 4.45	1.98 / 4.45	1.98 / 4.45	130 / 228	130 / 228	130 / 228	129 / 223	129 / 223	129 / 223
	6	5	60	yes	1.50 / 3.49	1.59 / 3.84	1.61 / 3.86	113 / 201	116 / 195	117 / 223	115 / 206	119 / 201	118 / 217
NSF	2	3	42	yes	2.04 / 5.78	1.89 / 5.29	2.00 / 5.32	139 / 272	128 / 253	137 / 279	133 / 279	134 / 266	134 / 292
	4	4	56	yes	1.56 / 4.01	1.77 / 4.57	1.53 / 4.04	119 / 220	114 / 200	120 / 220	119 / 207	141 / 266	117 / 220
	6	5	70	yes	1.34 / 2.90	1.48 / 3.81	1.47 / 3.31	110 / 192	108 / 233	121 / 200	113 / 192	124 / 233	112 / 200
Spain	1	3	42	no	7.24 / 21.24	7.24 / 21.24	7.24 / 21.24	240 / 480	240 / 480	240 / 480	237 / 472	237 / 472	237 / 472
	2	3	42	yes	2.61 / 7.45	3.18 / 9.39	2.82 / 7.85	155 / 323	170 / 354	163 / 323	145 / 315	161 / 409	148 / 275
	3	4	56	no	3.41 / 9.10	3.41 / 9.10	3.41 / 9.10	165 / 313	165 / 313	165 / 313	169 / 323	169 / 323	169 / 323
	4	4	56	yes	2.31 / 5.45	2.43 / 7.19	2.32 / 6.10	144 / 259	144 / 291	134 / 253	138 / 267	143 / 283	149 / 283
	5	5	70	no	3.02 / 7.30	3.02 / 7.30	3.02 / 7.30	155 / 293	155 / 293	155 / 293	162 / 283	162 / 283	162 / 283
	6	5	70	yes	2.17 / 5.53	2.12 / 6.19	2.10 / 4.90	133 / 267	128 / 251	130 / 251	139 / 259	140 / 275	139 / 251
Italia14	1	3	42	no	8.37 / 24.07	8.37 / 24.07	8.37 / 24.07	256 / 469	256 / 469	256 / 469	253 / 499	253 / 499	253 / 499
	2	3	42	yes	3.25 / 6.35	2.87 / 6.11	3.43 / 9.10	157 / 289	137 / 228	165 / 319	173 / 309	176 / 309	173 / 319
	3	4	56	no	4.42 / 12.36	4.42 / 12.36	4.42 / 12.36	189 / 359	189 / 359	189 / 359	189 / 349	189 / 349	189 / 349
	4	4	56	yes	2.80 / 5.86	2.81 / 6.11	2.82 / 6.95	154 / 268	162 / 299	150 / 268	153 / 268	146 / 238	156 / 289
	5	5	70	no	3.51 / 9.46	3.51 / 9.46	3.51 / 9.46	168 / 329	168 / 329	168 / 329	171 / 329	171 / 329	171 / 329
	6	5	70	yes	2.51 / 5.74	2.37 / 5.38	2.48 / 6.17	145 / 258	139 / 258	141 / 248	146 / 278	144 / 258	148 / 258
EPAN	1	3	48	no	8.48 / 28.13	8.48 / 28.13	8.48 / 28.13	269 / 561	269 / 561	269 / 561	249 / 612	249 / 612	249 / 612
	2	3	48	yes	3.15 / 8.34	3.61 / 12.55	2.80 / 6.71	160 / 348	166 / 369	156 / 285	172 / 341	182 / 362	158 / 271
	3	4	64	no	4.44 / 14.40	4.44 / 14.40	4.44 / 14.40	191 / 390	191 / 390	191 / 390	191 / 424	191 / 424	191 / 424
	4	4	64	yes	2.38 / 6.39	2.29 / 5.93	2.22 / 6.64	136 / 292	142 / 292	131 / 278	152 / 313	143 / 292	141 / 278
	5	5	80	no	3.62 / 8.79	3.62 / 8.79	3.62 / 8.79	167 / 320	167 / 320	167 / 320	176 / 355	176 / 355	176 / 355
	6	5	80	yes	2.24 / 5.61	2.19 / 5.38	2.22 / 5.73	138 / 285	136 / 264	138 / 250	139 / 292	140 / 271	138 / 285

Table 6.17: Traffic ratio scenarios.

# scenario	Traffic ratio	Class-1 demand	Class-3 demand	Class-4 demand
d1	50/0/50	1	0	1
d2	25/0/75	0.5	0	1.5
d3	5 / 0/95	0.1	0	1.9
d4	33/33/33	1	1	1
d5	17/42/42	0.5	1.25	1.25
d6	17/33/50	0.5	1	1.5
d7	17/50/33	0.5	1.5	1
d8	3/48/48	0.1	1.45	1.45
d9	3/17/80	0.1	0.5	2.4
d10	3/80/17	0.1	2.4	0.5

Finally, a conclusion similar to the model with protection given at lower layer can be made here with respect to the model with protection given at upper layer. Tables 6.15 and 6.16 show that the spine model stretches the upper bound of the availability of the 1+1 protection to offer a higher availability level with a manageable increase in resource use.

Thus far, we consider a unit of demand for each class, meaning that the percentage of the traffic of each class is 33% of the total demand. In reality, the majority of services (*e.g.*,

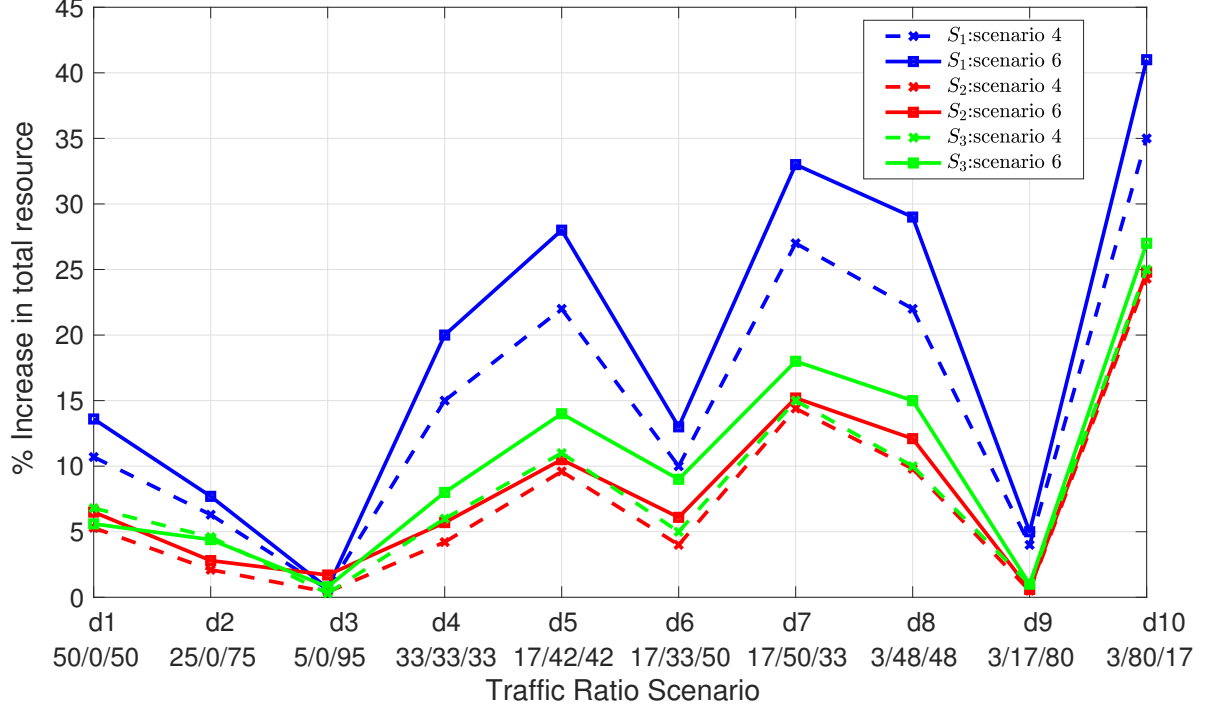


Figure 6.12: Percentage of increase in total resources compared to the baseline model in the Polska network and for different traffic ratio mix.

90% or more) do not require high availability levels. For this we consider varying the ratio of traffic between the classes to examine its impact on the design resource efficiency. Note that the above-mentioned design considers solving two problems separately (*i.e.*, the MILP in Section 6.4.3.1 and LP in Section 6.4.3.2). Thus it might not capture the impact of varying the traffic ratios between classes and to account for that we add the following constraints to the MILP in Section 6.4.3.1: Equations (6.23), (6.25), (6.33), (6.35) and (6.37) with $\phi = 3$, and Equations (6.24), (6.26), (6.34), (6.36) and (6.38) with $\phi = 4$. These constraints perform routing demands of class 3 and 4 on the logical layer, mapping them to the physical layer, and ensure that the mapping of each link is unique across different classes. We solved the optimization problem for scenario 4 and 6 in the Polska network considering additional scenarios with different traffic ratio mixes. Figure 6.12 shows the tested scenarios and the corresponding class traffic ratio and the demand for each class. We compare the resource use of the spine model to the no spine case (*i.e.*, the baseline model). Figure 6.12 shows

the percentage of increase in resources usage required by the spine model for each scenario. Note that the first three scenarios in Figure 6.12 consider only traffic of class 1 and 4, which are comparable to the scenarios of Table 6.8 with protection given at the lower layer. Their results are similar to the one observed in Figure 6.6 where the percentage of increase in total resources of the spine model decreases as class-1 traffic decreases. In the other cases d4 to d10, we observe that the resource efficiency of the spine model improves as the sum of class-1 and class-3 ratios decreases *i.e.*, trace d4, d6, and d9 scenarios.

6.5 SUMMARY

In this chapter, we introduced an approach that incorporates the spine and enables providing differentiated services over multilayer communications networks. The concept behind the spine is to create heterogeneous availability subnetworks at the physical layer to lay a basis for differentiation. Then, this differentiation capability is transferred to upper layers via crosslayer mapping to provide multiple logical networks with diverse availability levels. Two protection configurations were presented here. Our results showed that the spine model with protection given at upper layers can be more efficient in terms of resource usage compared to the spine model with protection given at the lower layer. Also based on creating two logical subnetworks in both models, the model with protection given at the upper can reuse each subnetwork to create an additional level of availability. However, the spine model with protection given at lower layer achieves slightly lower flows downtimes. In addition, our results show that both models can create a wider range of availability levels compared to existing techniques. The spine, in some cases, results into a slight increase in resource use. This however depends substantially on the layout of the logical layer, the spine used and the percentage of highest class traffic as well as the protection configuration.

7.0 CONCLUSIONS AND SUMMARY

This chapter summarizes the dissertation and discusses future work.

7.1 CONTRIBUTION

The objective of this dissertation was to study the problem of supporting differentiated classes of resilience in multilayer networks. Throughout this dissertation, we studied how to exploit heterogeneous link availability on a physical network to efficiently improve end-to-end availability, and therefore we explored the spine concept. Then we developed models that deploy the spine in order to provide different levels of availability to services of different classes of resilience in both single and multilayer networks. Here we summarize the content of this dissertation.

In Chapter 2, we provided a brief preliminary background on multilayer network design and how QoR classes are involved in such design. Subsequently, we surveyed the existing literature on supporting differentiated QoR classes in network design with a variety of design objectives in Chapter 3. The traditional way of achieving this differentiation considers alternate paths and/or protection schemes without altering network availability. We proposed to combine this approach with the spine concept in order to achieve higher availability values for the highest QoR class and create additional QoR classes using only basic protection schemes. This was presented in Chapter 4 where we stated our design questions and our methodology to address them.

In Chapter 5, we presented the spine concept of embedding a subgraph structure with higher availability in a network together with protection mechanisms. In general the spine is

hoped to provide larger differences in the range of availability values to quality of resilience classes resulting in less over engineering of the network to meet the most stringent availability requirements. Below we summarize our contributions and findings in this chapter.

1. Through a small network topology example, we explored the properties that impact the spine performance. The spine based approach was shown to have the potential to improve the network availability in a more efficient fashion compared to improving the availability of all network components in a homogeneous fashion. We showed that the efficiency of the spine is highly influenced by
 - a. The heterogeneity of link availability: the utility of the spine model is increased when the difference between the availability of the links on the spine and off the spine is significant. The more diverse the availability of the links, the better end-to-end availability.
 - b. The network size: the spine model can be more gainful to large networks and dense networks.
 - c. The spine layout: the spine model can be more gainful when the paths lengths on the spine are bounded. A star-like spine would have better end-to-end availability than a ring-like spine.
2. We provided heuristic spine selection methods based on structural properties of the network topology. The goal was to find a way to embed a spine to achieve a maximum average WP or end-to-end availability.
3. We compared the results of the heuristics to optimal spine values obtained by a brute force search, and the results appeared promising.
4. The optimal spines with respect to maximum average WP availability and maximum average end-to-end availability were found to have the following properties:
 - a. The optimal spine comes with relatively short diameters, small link betweenness and average hop count, and large link degree. This confirms that the spine is closer to star-like layout rather than ring.
 - b. The spine requires more bandwidth than a traditional shortest path-pairs approach. The spine with the optimal properties above requires the least increase in resources.

In most cases, the spine that maximizes average WP availability requires the minimum additional resources.

5. We provided an optimization based formulation for designing the spine taking into consideration that link availability is upgradeable at a given cost. The design problem aimed at exploiting existing heterogeneity in link availability and the upgradability of link availability to achieve a target flow availability value while minimizing the total cost. We found the following:

- a. The spine model exhibit superior efficiency in terms of average flow availability and potential advantage over the shortest path model with no spine.
- b. The efficiency of the spine depends primarily on the following:
 - The availability metric considered in the design *e.g.*, minimum flow WP availability and average flow availability.
 - The density of the network: the denser the network, the more efficient the spine.
 - The distribution of the cost of improving link availability:
 - The possibility of relaxing the MTTR of the off spine links, as such a relaxation can result in some saving in the total budget with only a slight impact on the end-to-end availability.
- c. The obtained spines have similar structural properties to the spines obtained from the heuristics with slight variations as the cost of improving link availability enforces an additional constraint in shaping the spine layout.

In Chapter 6, we presented our model that incorporates the spine and enables providing differentiated services over multilayer communications networks. The model considers a spine-aware crosslayer mapping to transfer the physical layer differentiation capability of the spine to the upper layer. Two protection configurations were presented, namely: protection at the lower layer and protection at the upper layer. Both configurations consider creating multiple logical subnetworks that are mapped on and off the spine and with or without protection so that logical paths of different QoR classes are distinguished. Further, we presented four possible QoR class arrangements for each protection configuration. We presented MILP formulations for each configuration considering creating two isolated logical subnetworks. Our results showed the following:

1. Both models can create a wider range of availability levels compared to basic protection schemes 0:1 and 1+1.
2. In both models, the spine results in a slight increase in resource use. This increase can be managed by optimizing the layout of the logical layer network, the choice of spine layout, and integrating accurate traffic demand estimates for each class into the design problem.
3. The spine model with protection given at upper layers can be more efficient in terms of resource usage compared to the spine model with protection given at the lower layer. Also based on creating only two logical subnetworks in both models, the former model can reuse each subnetwork to create an additional level of availability while improving the resource efficiency. In addition, comparing the resource efficiency of each spine model to equivalent models with no spine showed that the both spine models are equivalent.
4. Both spine models achieve small average flow downtimes compared to the no spine model, with slightly lower flows downtimes when protection is given at lower layer.

7.2 LIMITATIONS AND FUTURE RESEARCH

- **Spine design targeting a minimum end-to-end flow availability:**

In Section 5.5.1, we presented the spine link selection design problem. The problem was formulated as an MILP optimization problem that minimizes the total cost for designing the spine while meeting minimum availability targets for flows WP and BP separately. Even though setting a high availability target for WP availability or setting both WP and BP availability targets can result in a spine with efficiently large end-to-end flow availability, this does not necessarily correspond to setting a target availability value for the minimum (or average) end-to-end availability (*i.e.*, $\min_f A_f, A_S$). Nevertheless, modifying our optimization problem to ensure a target end-to-end flow availability would give rise to inevitable nonlinearity – if implemented straightforwardly. Hence developing a heuristics method to design the spine targeting a minimum end-to-end flow availability

–as well as any other metric of interest– would be an extension that complements this work.

- **Steiner tree Spine:**

The spine was designed to be a spanning tree to ensure high availability and availability differentiation between all node pairs in the network. However, it is likely to have a case where the spine is needed to support communications between only a subset of nodes in the network (*i.e.*, a subset of the full-mesh). In this case, we would have a Steiner tree spine instead of a spanning tree, where the spine spans only the nodes that are spine nodes and may include nodes that are other intermediate nodes that are not part of the supported node pairs subset. We expect this problem to be harder but is worth investigating. Actually one can consider that the spine problem presented in this dissertation a special case of the Steiner tree spine problem.

- **Node availabilities:**

In addition to the previous point related to the design of the spine, we also assumed that nodes are perfect since, as reported, they have better availability than links. However, we would expect that if this was not the case or there was considerable heterogeneity in nodes availabilities, node availability could have an impact on the spine design, and more specifically, in the following scenarios: First, in the Steiner tree spine problem in which intermediate nodes that are not part of the high availability communications would have different availabilities and the selection of such a node impacts the supported flows availabilities. Second, the consideration of the node-disjointness requirement for a path-pair besides the link-disjointness requirement. Here we expect that considering node availability would be a helpful basis for altering the disjointness requirement for each flow, if applicable, and partially tuning its overall availability. In general, the consideration of node availability is complementary to the whole design.

- **Logical network design:**

Typically, the problem of communications network design is decomposed into multiple subproblems to reduce the complexity or to evaluate a specific aspect of the design solely while neutralizing other variables. In the design of the multilayer spine model in Chapter 6, we assumed that the spine and the logical network are given. Our analysis

showed a great dependency of the spine efficiency on both the spine layout and the layout of the logical network. One way to improve this work is to integrate the logical spine design and/or the spine layout into the multilayer spine design of Chapter 6.

- **Capacity and demand considerations:**

Even though our models considered the volume of demands between node pairs, physical and logical links capacities, and the aggregated bandwidth/capacity in each layer, we actually used uniform units of demand for all flows and relaxed the capacity constraints in our numerical study in order to evaluate the impact and the performance of the spine solely, removing any potential bias enforced by other constraints. This approach is not uncommon in communication network design research. However, it is important to consider scenarios in which demand estimates and link capacities are included, and study their impact on the whole design. Specifically, one can study the impact of different traffic demands distribution on the optimal spine design, and to what extent an optimal spine layout is robust against demand variations.

- **Additional protection schemes:**

In Chapter 5, we used a single protection scheme 1+1. One possible extension to our work is to consider creating additional QoR classes by incorporating more complex protection schemes (*e.g.*, 1+N, multipath routing, link restoration. See section 3.2.1) at lower or upper layer.

- **Reduce complexity:**

The proposed design problems in Chapters 5 and 6 are NP-complete. This means that solving these optimization problems for large networks or for some instances might be intractable. Thus an immediate extension for our research is to develop computationally efficient heuristic algorithms for these problems with acceptable optimality level.

- **Statistical accuracy:**

In Chapter 6, we formulated spine model in multilayer networks with two protection configurations. We studied a number of performance aspects for each configuration using 7 random networks (*i.e.*, 7 repetitions) and the results of interest were averaged. Some of these results were statistically weak. For example, in Table 6.9 the confidence interval around the averaged resource values in some cases reaches up to $\pm 10.8\%$ with a 95%

confidence level and up to $\pm 30.2\%$ with 95% confidence around the average downtime values in Table 6.13. Hence our numerical studies can be solidified by considering extra repetitions of the random logical networks when needed. Note that the findings build on these results can be considered expressive as we noticed the same pattern of behavior across different networks, but this might not always be the case.

- **Application to Software Defined Networking (SDN):**

Software Defined Networking (SDN) technologies are likely to play an important role in future networks. SDN is based on a centralized control architecture that maintains a global view of network status and performs network management functions. A single controller or logically centralized group of controllers can be deployed to maintain these functions. The centralized architecture, however, introduces a single point of failure, and the availability of the control paths (*i.e.*, connecting controllers to forwarding devices) becomes crucial to the whole system. On this context, one can study how the SDN design can benefit from the spine model. Specifically two problems are suggested; 1) how to design the spine in order to maximize the availability of control paths from a single or multiple controllers to all switches, 2) restate the controller placement problem taking into consideration that a spine is already deployed in the network.

- **Economics of the Spine** In Section 5.5.5, we identified several scenarios where the spine model is efficient in terms of the design cost and flow availability. The findings demonstrate that the spine is a promising and feasible option for supporting differentiated classes of resilience. Though, it is crucial to study the spine concept approach economically from a wider perspective in a fashion similar to [135, 136]. A study that complements our evaluation of the spine model might consider the following problem inputs:

1. A customized spine cost model that estimates capital expenditures (CAPEX) and operational expenditures (OPEX) invested to embed the spine in the network. The CAPEX costs represent the cost of purchasing and installing new equipment (*e.g.*, installations costs for improving MTBF of a cable), and the OPEX costs here are mainly related to maintenance work. The model should also consider the distribution of these costs over the planning interval of the investment.

2. The expected volume of traffic that will use the spine (includes both unprotected and protected traffic) over the investment interval.
3. The projected profit sought from investing in the spine (*i.e.*, the desired Return on investment, ROI)

Then, the pricing rate for each class of traffic can be assessed ensuring profitable investment from the service provider perspective. In addition, including an equivalent study for an alternative approach (*e.g.*, based on high order protection configuration or improving all network availability) would also solidify the conclusion. From a marketing perspective, a comparative analysis of the pricing rates of the spine and the alternative approaches is also required to evaluate the spine approach economically.

BIBLIOGRAPHY

- [1] T. G. Lewis, *Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation*. Wiley-Interscience, Apr. 2006.
- [2] J. Tapolcai, P. Cholda, T. Cinkler, K. Wajda, A. Jajszczyk, A. Autenrieth, S. Bodamer, D. Colle, G. Ferraris, H. Lonsethagen, I. E. Svinnet, and D. Verchere, “Quality of resilience (QoR): NOBEL approach to the multi-service resilience characterization,” in *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, Oct 2005, pp. 1328–1337 Vol. 2.
- [3] U. Franke, R. Lagerström, M. Ekstedt, J. Saat, and R. Winter, “Trends in enterprise architecture practice—a survey,” in *TrendsEnterprise Architecture Research Workshop*., 2010.
- [4] M. Rausand and R. Hoyland, *System Reliability Theory: Models, Statistical Methods and Applications*, 2ed, Ed. Wiley Interscience, 2003.
- [5] P. Cholda, E. L. Følstad, B. E. Helvik, P. Kuusela, M. Naldi, and I. Norros, “Towards risk-aware communications networking,” *Reliability Engineering and System Safety*, vol. 109, pp. 160–174, 2013.
- [6] *Communications Requirements of Smart Grid Technologies*, US Department of Energy, 2010.
- [7] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann Publishers, Elsevier, 2004.
- [8] E. Marcus and H. Stern, *Blueprints for High Availability*, 2nd ed. Wiley Publishing, 2003.
- [9] D. A. Garbin, J. E. Knepley, F. Park, D. South, and F. Church, “Design and Analysis of High Availability Networks,” in *IEEE Conference on Technologies for Homeland Security*, 2009, pp. 1–6.

- [10] P. Cholda, A. Mykkeltveit, B. Helvik, O. Wittner, and A. Jajszczyk, "A Survey Of Resilience Differentiaion Frameworks in Communication Networks," *IEEE Communications Surveys*, vol. 9, no. 4, pp. 32–55, 2007.
- [11] H. Zhang and A. Durresi, "Differentiated multi-layer survivability in IP/WDM networks," *NOMS 2002. IEEE/IFIP Network Operations and Management Symposium. 'Management Solutions for the New Communications World'(Cat. No.02CH37327)*, pp. 681–694, 2002.
- [12] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger, "General availability model for multilayer transport networks," in *Design of Reliable Communication Networks, 2005. DRCN 2005. 5th IEEE International Workshop on*, 2005.
- [13] P. Pacharintanakul and D. Tipper, "Crosslayer survivable mapping in overlay-IP-WDM networks," in *Proceedings of the 2009 7th International Workshop on the Design of Reliable Communication Networks, DRCN 2009*. IEEE, 2009, pp. 168–174.
- [14] D. Lai, W. and McDysan, "RFC 3386, Network Hierarchy and Multilayer Survivability," 2002.
- [15] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks: A Practical Perspective, 3rd Edition*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009.
- [16] V. Sharma and F. Hellstrand, "Framework for Multi-Protocol Label Switching (MPLS)-based Recovery," IETF RFC 3469, 2003.
- [17] W. Bigos, B. Cousin, S. Gosselin, M. Le Foll, and H. Nakajima, "Survivable MPLS over optical transport networks: Cost and resource usage analysis," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 5, pp. 949–962, 2007.
- [18] T. Lin, Z. Zhou, K. Thulasiraman, G. Xue, and S. Sahni, "Unified Mathematical Programming Frameworks for Survivable Logical Topology Routing in IP-over-WDM Optical Networks," *Journal of Optical Communications and Networking*, vol. 6, no. 2, pp. 190–203, 2014.
- [19] Z. Zhou, T. Lin, K. Thulasiraman, G. Xue, and S. Sahni, "Novel survivable logical topology routing in IP-over-WDM networks by logical protecting spanning tree set," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, 2012, pp. 650–656.
- [20] W. Molisz and J. Rak, *Computer Network Security: 5th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2010, St. Petersburg, Russia, September 8-10, 2010. Proceedings*. Springer Berlin, Heidelberg, 2010, ch. A Novel Genetic Approach to Provide Differentiated Levels of Service Resilience in IP-MPLS/WDM Networks, pp. 307–320. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-14706-7_24

- [21] J. Rak, M. Pickavet, K. S. Trivedi, J. A. Lopez, A. M. C. A. Koster, J. P. G. Sterbenz, E. K. Cetinkaya, T. Gomes, M. Gunkel, K. Walkowiak, and D. Staessens, "Future research directions in design of reliable communication systems," *Telecommunication Systems*, vol. 60, no. 4, pp. 423–450, 2015. [Online]. Available: <http://dx.doi.org/10.1007/s11235-015-9987-7>
- [22] ITU-T, "Recommendation E.800: Quality of service and dependability vocabulary," 1994.
- [23] E. Bauer, R. Adams, and D. Eustace, *Beyond Redundancy: How Geographic Redundancy can Improve Service Availability and Reliability of Computer-Based Systems*, 1st ed. Wiley-IEEE Press, 2011.
- [24] D. Mello, D. Schupke, M. Scheffel, and H. Waldman, "Availability maps for connections in WDM optical networks," in *DRCN 2005. 5th International Workshop on Design of Reliable Communication Networks - "Reliable Networks for Reliable Services"*. IEEE, 2005, pp. 77–84. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1563847>
- [25] M. L. Shooman, *Reliability of Computer Systems and Networks*. New York, USA: John Wiley & Sons, Inc., jan 2002, vol. 107. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/047122460X.fmatter{ }indsub/summaryhttp://doi.wiley.com/10.1002/047122460X>
- [26] A. J. Gonzalez and B. E. Helvik, "SLA success probability assessment in networks with correlated failures," *Computer Communications*, vol. 36, no. 6, pp. 708–717, Mar. 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0140366412002708>
- [27] A. P. Snow and G. R. Weckman, "What Are the Chances an Availability SLA will be Violated?" in *Sixth International Conference on Networking (ICN'07)*. IEEE, apr 2007, pp. 35–35. [Online]. Available: <http://ieeexplore.ieee.org/document/4196228/>
- [28] P. Cholda, A. Mykkeltveit, B. E. Helvik, and A. Jajszczyk, "Continuity-based resilient communication," *Proceedings of the 2009 7th International Workshop on the Design of Reliable Communication Networks, DRCN 2009*, pp. 335–342, 2009.
- [29] B. Statovci-Halimi and G. Franzl, "QoS differentiation and Internet neutrality," *Telecommunication Systems*, vol. 52, no. 3, pp. 1605–1614, mar 2013. [Online]. Available: <http://link.springer.com/10.1007/s11235-011-9517-1>
- [30] S. Awerbuch and A. Preston, Eds., *The Virtual Utility*. Boston, MA: Springer US, 1997. [Online]. Available: <http://link.springer.com/10.1007/978-1-4615-6167-5>
- [31] B. A. Cherry, *The Crisis in Telecommunications Carrier Liability*. Boston, MA: Springer US, 1999. [Online]. Available: <http://link.springer.com/10.1007/978-1-4615-4993-2>

- [32] D. Khader, J. Padget, and M. Warnier, *Reactive Monitoring of Service Level Agreements*. Boston, MA: Springer US, 2010, pp. 13–22. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-7320-7_2
- [33] M. To and P. Neusy, “Unavailability analysis of long-haul networks,” *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 1, pp. 100–109, January 1994.
- [34] F. Lawler. (2011, August) The 10 most bizarre and annoying causes of fiber cuts. [Online]. Available: <http://blog.level3.com/level-3-network/the-10-most-bizarre-and-annoying-causes-of-fiber-cuts/>
- [35] NRSC, “Network Reliability Steering Committee (NRSC), Annual Report 2004,” ATIS, Tech. Rep., 2005.
- [36] A. P. Snow, “A reliability assessment of the public switched telephone network infrastructure,” Ph.D. dissertation, School of Information Sciences, Pittsburgh, PA, USA, 1997, aAI9812414.
- [37] W. D. Grover, “Failure Impacts, Survivability Principles, and Measures of Survivability,” in *Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*, W. D. Grover, Ed. Upper Saddle River, New Jersey: Prentice Hall PTR, 2003, ch. 3, pp. 103–172. [Online]. Available: <http://www.ece.ualberta.ca/~grover/book/>
- [38] D. Crawford, “Fiber optic cable dig-ups: Causes and cures,” 1993.
- [39] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. N. Chuah, Y. Ganjali, and C. Diot, “Characterization of failures in an IP backbone,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 749–762, 2008.
- [40] B. De La Cruz, O. Gonzalez De Dios, V. Lopez, and J. P. Fernandez-Palacios, “Operational expenditures savings in IP/MPLS over DWDM networks by Multi-layer restoration,” in *Conference on Optical Fiber Communication, Technical Digest Series*. Optical Society of America, 2014, pp. 5–7.
- [41] I. Rados, T. Sunaric, and P. Turalija, “Suggestions for Availability Improvement of Optical Cables,” in *Circuits and Systems for Communications, 2002. Proceedings. ICCSC '02. 1st IEEE International Conference on*. IEEE, 2002, pp. 234–239.
- [42] H. C. Cankaya, A. Lardies, and G. W. Ester, “Availability Aware Cost Modeling of Mesh Architectures for Long-Haul Networks,” in *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on*, vol. 2, 2004, pp. 766–771.
- [43] NRSC and ATIS, “Alliance for Telecommunications Industry Solutions (ATIS) Network Reliability Steering Committee (NRSC) Bulletin No. 2014-00, Planned Maintenance Related Outages,” ATIS, Tech. Rep., 2014.

- [44] NRSC and FCC, “The Network Reliability Steering Committee, Federal Communications Commission, Bulletin No. 2009-008 E-911 Outages,” NRSC, Tech. Rep., 2009.
- [45] H.-Y. Chang, I.-K. Chen, and W.-C. Chiang, “A Simple Network-Engineering Approach for Improving Connection Availability in WDM Networks,” *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, pp. 805–807, Mar. 2011.
- [46] W. D. Grover and A. Sack, “High availability survivable networks: When is reducing mtrr better than adding protection capacity?” *6th International Workshop on Design and Reliable Communication Networks.*, 2007.
- [47] J. Chen, L. Wosinska, C. Machuca, and M. Jaeger, “Cost vs. reliability performance study of fiber access network architectures,” *IEEE Communications Magazine*, pp. 56–65, February 2010.
- [48] “Reliability of fiber optic cable systems: buried fiber optic cable optical groundwire cable all dielectric, self supporting cable,” ALCOA FUJIKURA LTD, Tech. Rep. May, 2001. [Online]. Available: <http://www.southern-telecom.com/solutions/afl-reliability.pdf>
- [49] J. Zhang, E. Modiano, and D. Hay, “Enhancing network robustness via shielding,” in *Design of Reliable Communication Networks (DRCN), 2015 11th International Conference on the*, March 2015, pp. 17–24.
- [50] P. E. Heegaard and K. S. Trivedi, “Network survivability modeling,” *Computer Networks*, vol. 53, no. 8, pp. 1215–1234, Jun. 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1389128609000425>
- [51] P. Cholda and A. Jajszczyk, “Recovery and Its Quality in Multilayer Networks,” *Journal of Lightwave Technology*, vol. 28, no. 4, pp. 372–389, 2010.
- [52] W. Ramirez, X. Masip-Bruin, E. Marin-Tordera, and S. Sánchez-López, “Managing resilience in carrier grade networks: Survey, open issues and trends,” *Computer Communications*, vol. 61, no. 0, pp. 1 – 16, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S014036641500095X>
- [53] F. Gont, R. Atkinson, and C. Pignataro, “Recommendations on Filtering of IPv4 Packets Containing IPv4 Options,” IETF RFC 7126, feb 2014. [Online]. Available: <https://www.rfc-editor.org/info/rfc7126>
- [54] L. Wosinska, D. Colle, P. Demeester, K. Katrinis, M. Lackovic, O. Lapcevic, I. Lievens, G. Markidis, B. Mikac, M. Pickavet, B. Puype, D. Staessens, and A. Tzanakaki, *Network Resilience in Future Optical Networks*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5412, ch. 9, pp. 253–284.

- [55] D. Harle, S. Albarrak, F. Ali, A. Urria, E. Calle, and J. L. Marzo, "Service level agreement framework for differentiated survivability in GMPLS-based IP-over-optical networks," in *IEEE International Conference on Communications*, 2007, pp. 2249–2256.
- [56] J. Zhang, S. Member, K. Zhu, H. Zang, and N. S. Matloff, "Availability-Aware Provisioning Strategies for Differentiated Protection Services in Wavelength-Convertible WDM Mesh Networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1177–1190, 2007.
- [57] R. Ramamurthy and B. Mukherjee, "Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks," *{IEEE/ACM} Transactions on Networking*, vol. 10, no. 3, pp. 351–367, Jun. 2002.
- [58] L. Song and B. Mukherjee, "New Approaches for Dynamic Routing with Availability Guarantee for Differentiated Services in Survivable Mesh Networks: The Roles of Primary-Backup Link Sharing," in *IEEE GLOBECOM 2006*, 2006. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4151029
- [59] Q. She, X. Huang, and J. P. Jue, "How Reliable Can Two-Path Protection Be?" *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 922–933, 2010.
- [60] M. Tornatore, D. Lucerna, B. Mukherjee, and A. Pattavina, "Multilayer Protection with Availability Guarantees in Optical WDM Networks," *Journal of Network and Systems Management*, vol. 20, no. 1, pp. 34–55, Sep. 2012.
- [61] A. Nafarieh, S. C. Sivakumar, W. Phillips, and W. Robertson, "Memory-aware SLA-based mechanism for shared-mesh WDM networks," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, Oct 2011, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6078998
- [62] L. Guo, H. Yu, and L. Li, "Joint routing-selection algorithm for a shared path with differentiated reliability in survivable wavelength-division-multiplexing mesh networks." *Optics express*, vol. 12, no. 11, pp. 2327–2337, 2004.
- [63] D. Schupke, "Guaranteeing service availability in optical network design," *Proc. SPIE 6022, Network Architectures, Management, and Applications*, vol. 6022, 2005. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=721548>
- [64] R. He, B. Lin, and L. Li, "Dynamic service-level-agreement aware shared-path protection in WDM mesh networks," *Journal of Network and Computer Applications*, vol. 30, no. 2, pp. 429–444, Apr. 2007. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1084804506000476>
- [65] L. Song, J. Zhang, and B. Mukherjee, "Dynamic provisioning with availability guarantee for differentiated services in survivable mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 35–43, 2007.

- [66] B. Kantarci, H. Mouftah, and S. Oktug, "Connection provisioning with feasible shareability determination for availability-aware design of optical networks," in *Proceedings of 2008 10th Anniversary International Conference on Transparent Optical Networks, ICTON*, no. 2214, 2008, pp. 19–22. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4598645
- [67] S. Yang, S. Trajanovski, and F. Kuipers, "Availability-Based Path Selection," in *Proc. Of the 6th International Workshop on Reliable Networks Design and Modeling (RNDM2014)*, Barcelona, Spain, 2014, pp. 17–19. [Online]. Available: <http://www.nas.ewi.tudelft.nl/people/Fernando/papers/RNDM2014.pdf>
- [68] H. Lee, E. Modiano, and K. Lee, "Diverse routing in networks with probabilistic failures," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 6, pp. 1895 – 1907, May 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5473148
- [69] P. Babarczi and J. Tapolcai, "End-to-End Service Availability Guarantee with Generalized Dedicated Protection," *Proceedings of the 6th International Symposium Communication Systems, Networks and Digital Signal Processing, CSNDSP 08*, pp. 511–515, 2008.
- [70] T. Miyamura, T. Kurimoto, A. Misawa, and S. Urushidani, "A disjoint path selection scheme based on enhanced shared risk link group management for multi-reliability service," in *GLOBECOM - IEEE Global Telecommunications Conference*, vol. 4. IEEE, 2005, pp. 1879–1884.
- [71] D. A. A. Mello, J. U. Pelegri, R. P. Ribeiro, D. A. Schupket, and H. Waldman, "Dynamic provisioning of shared-backup path protected connections with guaranteed availability requirements," in *2nd International Conference on Broadband Networks, BROADNETS 2005*, vol. 2005, 2005, pp. 397–404. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1589761
- [72] N. H. Bao, L. M. Li, H. B. Luo, Z. Z. Zhang, and H. F. Yu, "On exploiting sharable resources with resource contention resolution for surviving double-link failures in optical mesh networks," *Journal of Lightwave Technology*, vol. 30, no. 17, pp. 2788–2795, 2012.
- [73] M. M. A. Azim and M. N. Kabir, "Availability study of M:N automatic protection switching scheme in WDM networks," *High Speed Networks*, vol. 18, pp. 1–13, 2012.
- [74] W. Fawaz, F. Martignon, K. Chen, and G. Pujolle, "A priority-aware protection technique for quality of service enabled WDM networks," *Lecture Notes in Computer Science*, vol. 3462, pp. 419–430, 2005.
- [75] H. Alshaer and J. M. H. Elmirghani, "Probabilistic differentiated availability services in WDM networks protected by shared protection schemes," *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management, IM 2011*, pp. 281–288, 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5990702

- [76] P. Demeester, M. Gryseels, A. Autenrieth, C. Brianza, L. Castagna, G. Signorelli, R. Clemente, M. Ravera, A. Jajszczyk, D. Janukowicz, K. V. Doorselaere, and Y. Harada, "Resilience in multilayer networks," *IEEE Communications Magazine*, vol. 37, no. 8, pp. 70–76, August 1999.
- [77] A. Mykkeltveit and B. E. Helvik, "On provision of availability guarantees using shared protection," *ONDM 2008 - 12th Conference on Optical Network Design and Modelling*, 2008.
- [78] R. Doverspike and B. Cortez, "Restoration in carrier networks," *Proceedings of the 2009 7th International Workshop on the Design of Reliable Communication Networks, DRCN 2009*, pp. 45–54, 2009.
- [79] C. Diot, G. Iannaccone, A. Markopoulou, C.-n. Chuah, and S. Bhattacharyya, "Service Availability in IP Networks," Sprint ATL Research, Tech. Rep. RR03-ATL-071888, 2003.
- [80] L. Zhou, M. Held, A. Member, and U. Sennhauser, "Connection Availability Analysis of Shared Backup Path-Protected Mesh Networks," *Journal of Lightwave Technology*, vol. 25, no. 5, pp. 1111–1119, 2007.
- [81] S. Huang, C. Martel, and B. Mukherjee, "Adaptive Reliable Multipath Provisioning in Survivable WDM Mesh Networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 6, pp. 368–380, 2010.
- [82] H. Y. Chang, "A multipath routing algorithm for degraded-bandwidth services under availability constraint in WDM networks," in *Proceedings - 26th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2012*. IEEE, Mar. 2012, pp. 881–884. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6185357>
- [83] H. Ma, D. Fayek, and P. H. Ho, "Availability-constrained multipath protection in backbone networks with double-link failure," in *IEEE International Conference on Communications*. IEEE, 2008, pp. 158–164. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4533073>
- [84] M. Clouqueur and W. D. Grover, "Availability Analysis and Enhanced Availability Design in p-Cycle-Based Networks," *Photonic Network Communications*, pp. 55–71, 2005.
- [85] D. S. Mukherjee, C. Assi, and A. Agarwal, "An Alternative Approach for Enhanced Availability Analysis and Design Methods in p-Cycle-Based Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 1–12, 2006.
- [86] S. Sebbah and B. Jaumard, "Differentiated quality-of-protection in survivable WDM mesh networks using p-structures," *Computer Communications*, vol. 36, no. 6, pp. 621–629, 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.comcom.2012.09.003>

- [87] R. Lin, S. Wang, L. Li, and L. Guo, "A new network availability algorithm for WDM optical networks," in *The Fifth International Conference on Computer and Information Technology (CIT'05)*. IEEE, 2005. [Online]. Available: <http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.ieee-000001562697>
- [88] B. Kantarci, H. Mouftah, and S. Oktug, "Adaptive schemes for differentiated availability-aware connection provisioning in optical transport networks," *Journal of Lightwave Technology*, vol. 27, no. 20, pp. 4595–4602, 2009. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5075664
- [89] Q. Guo, P. H. Ho, A. Haque, and H. T. Mouftah, "Availability-constrained Shared Backup Path Protection (SBPP) for GMPLS-based spare capacity reprovisioning," in *IEEE International Conference on Communications*, 2007, pp. 2186–2191.
- [90] D. Lucerna, M. Tornatore, B. Mukherjee, and A. Pattavina, "Trading availability among shared-protected dynamic connections in WDM networks," *Computer Networks*, vol. 56, no. 13, pp. 3150–3162, Sep. 2012. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1389128612001612>
- [91] O. Gerstel and G. Sasaki, "Meeting SLAs by design: A protection scheme with memory," in *OFC/NFOEC 2007 - Optical Fiber Communication and the National Fiber Optic Engineers Conference*, 2007.
- [92] L. S. L. Song and B. Mukherjee, "Accumulated-Downtime-Oriented Restoration Strategy With Service Differentiation in Survivable WDM Mesh Networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 1, no. 1, pp. 113–124, 2009.
- [93] S. Kamamura, T. Miyamura, and K. Shiimoto, "Relaxed maintenance protection architecture by dynamic backup path configuration," in *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference*. Washington, D.C.: OSA, 2011, pp. 1–3.
- [94] J. Ahmed and C. Cavdar, "Hybrid survivability schemes achieving high connection availability with a reduced amount of backup resources," *Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. 152–161, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6645109
- [95] A. Martinez, M. Yannuzzi, V. Lopez, D. Lopez, W. Ramirez, R. Serral-Gracia, X. Masip-Bruin, M. Maciejewski, and J. Altmann, "Network management challenges and trends in multi-layer and multi-vendor settings for carrier-grade networks," *IEEE Communications Surveys & Tutorials*, pp. 1–24, 2014. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6826469>
- [96] K. Vajanapoom, D. Tipper, and S. Akavipat, "Risk based resilient network design," *Telecommunication Systems*, vol. 52, pp. 799–811, Aug. 2013.

- [97] H.-W. Lee, K. Lee, and E. Modiano, “Maximizing Reliability in WDM Networks Through Lightpath Routing,” *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1052–1066, Aug. 2014.
- [98] P. Pacharintanakul and D. Tipper, “Link availability mapping in infrastructure-based overlay networks,” in *2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, ICUMT 2010*, 2010, pp. 552–558.
- [99] W. Ni, J. Wu, C. Huang, and M. Savoie, “Analytical models of flow availability in two-layer networks with dedicated path protection,” *Optical Switching and Networking*, vol. 10, no. 1, pp. 62–76, Jan. 2013. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1573427712000707>
- [100] P. Hegyi, M. Maliosz, A. Ladányi, and T. Cinkler, “Virtual Private/Overlay Network Design With Traffic Concentration and Shared Protection,” *Journal of Network and Systems Management*, vol. 13, no. 1, pp. 119–138, Mar. 2005. [Online]. Available: <http://link.springer.com/10.1007/s10922-005-1865-2>
- [101] S. Herker, X. An, W. Kiess, and A. Kirstadter, “Path protection with explicit availability constraints for virtual network embedding,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2013, pp. 2978–2983. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6666657
- [102] Q. Zheng, S. Member, G. Cao, and T. F. L. Porta, “Cross-Layer Approach for Minimizing Routing Disruption in IP Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1659–1669, 2014.
- [103] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, “Resilient overlay networks,” *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 1, p. 66, Jan. 2002. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=510726.510740>
- [104] J. Kurian and K. Sarac, “A survey on the design, applications, and enhancements of application-layer overlay networks,” *ACM Computing Surveys*, vol. 43, no. 1, pp. 1–34, Nov. 2010. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1824795.1824800>
- [105] Z. Duan, Z. L. Zhang, and Y. T. Hou, “Service overlay networks: SLAs, QoS and bandwidth provisioning,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 334–343, 2003.
- [106] Z. Li and P. Mohapatra, “On investigating overlay service topologies,” *Computer Networks*, vol. 51, no. March 2006, pp. 54–68, 2007.
- [107] H. L. H. Lee and J. K. J. Kim, “A Service Availability-aware Construction of Profitable Service Overlay Network,” *2009 11th International Conference on Advanced Communication Technology*, vol. 02, pp. 1252–1256, 2009.

- [108] V. Miletic, “Method for Optimizing Availability of Optical Telecommunication Network in Presence of Correlated Failures,” Ph.D. dissertation, University of Zagreb, 2015.
- [109] T. Gomes, D. Tipper, and A. Alashaikh, “A novel approach for ensuring high end-to-end availability: The spine concept,” in *Design of Reliable Communication Networks (DRCN), 2014 10th International Conference*. IEEE, 2014, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/6816142/>
- [110] Q. Botton, B. Fortz, and L. Gouveia, “On the hop-constrained survivable network design problem with reliable edges,” *Computers & Operations Research*, vol. 64, pp. 159–167, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0305054815001264>
- [111] A. Alashaikh, T. Gomes, and D. Tipper, “The spine concept for improving network availability,” *Computer Networks*, vol. 82, pp. 4–19, May 2015.
- [112] A. Alashaikh, D. Tipper, and T. Gomes, “Supporting differentiated resilience classes in multilayer networks,” in *2016 12th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, mar 2016, pp. 31–38. [Online]. Available: <http://ieeexplore.ieee.org/document/7470832/>
- [113] D. Tipper, “Resilient network design: Challenges and future directions,” *Telecommunication Systems*, vol. 56, no. 1, pp. 5–16, 2014.
- [114] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956.
- [115] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, “SNDlib 1.0–Survivable Network Design library,” *Networks*, vol. 55, no. 3, pp. 276–286, 2010.
- [116] R. Martínez, R. Casellas, R. Vilalta, and R. Muñoz, “GMPLS / PCE-Controlled Multi-Flow Optical Transponders in Elastic Optical Networks [Invited],” *Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. 71–80, 2015.
- [117] D. Colle, S. De Maesschalck, C. Develder, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, and P. Demeester, “Data-centric optical networks and their survivability,” *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 6–20, 2002.
- [118] Sophie De Maesschalck, Didier Colle, Ilse Lievens, Mario Pickavet, Piet Demeester, Christian Mauz, Monika Jaeger, Robert Inkret, Branko Mikac, and Jan Derkacz, “Pan-European Optical Transport Networks: An Availability-Based Comparison,” *Photonic Network Communications*, vol. 5, no. 3, pp. 203–225, May 2003.

- [119] M. Tornatore, G. Maier, and A. Pattavina, “Availability Design of Optical Transport Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 8, pp. 1520–1532, 2005.
- [120] G. R. Chris Godsil, *Algebraic Graph Theory*. Springer, 2001.
- [121] F. Piedad and M. Hawkins, *High Availability: Design, Techniques, and Processes*. Prentice Hall, 2001.
- [122] S. Herker, W. Kiess, X. An, and A. Kirstadter, “On the trade-off between cost and availability of virtual networks,” *2014 IFIP Networking Conference, IFIP Networking 2014*, 2014.
- [123] U. Franke, “Optimal IT service availability: Shorter outages, or fewer?” *IEEE Transactions on Network and Service Management*, vol. 1, no. 9, pp. 22–33, 2012.
- [124] R. M. Karp, “Reducibility among combinatorial problems,” in *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 219–241.
- [125] J. Q. Hu, “Diverse routing in optical mesh networks,” *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 489–494, March 2003.
- [126] C.-L. Li, S. Thomas McCormick, and D. Simchi-Levi, “Finding disjoint paths with different path-costs: Complexity and algorithms,” *Networks*, vol. 22, no. 7, pp. 653–667, dec 1992. [Online]. Available: <http://dx.doi.org/10.1002/net.3230220705><http://doi.wiley.com/10.1002/net.3230220705>
- [127] T. Nagae and H. Wakabayashi, “Differences in Network Reliability Improvement by Several Importance Indices,” *Transportation Research Procedia*, vol. 10, pp. 155–165, 2015. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S2352146515002525>
- [128] K. Lee, H.-w. Lee, and E. Modiano, “Reliability in Layered Networks with Random Link Failures,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1835–1848, 2011.
- [129] E. Dolan, “The NEOS server 4.0 administrative guide,” Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory, Tech. Rep., 2001.
- [130] W. Gropp and J. J. Moré, “Optimization environments and the NEOS server,” in *Approximation Theory and Optimization*, M. D. Buhmann and A. Iserles, Eds. Cambridge University Press, 1997, pp. 167–182.
- [131] J. Czyzyk, M. P. Mesnier, and J. J. More, “The NEOS server,” *IEEE Journal on Computational Science and Engineering*, vol. 5, no. 3, pp. 68–75, 1998.

- [132] F. Kuipers and F. Dijkstra, "Path selection in multi-layer networks," *Computer Communications*, vol. 32, no. 1, pp. 78–85, jan 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0140366408005124>
- [133] M. L. Lamali, N. Fergani, J. Cohen, and H. Pouyllau, "Path computation in multi-layer networks: Complexity and algorithms," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. IEEE, apr 2016, pp. 1–9. [Online]. Available: <http://arxiv.org/abs/1601.01786><http://dx.doi.org/10.1109/INFOCOM.2016.7524550><http://ieeexplore.ieee.org/document/7524550/>
- [134] F. A. Kuipers and P. V. Mieghem, "The impact of correlated link weights on QoS routing," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, March 2003, pp. 1425–1434 vol.2.
- [135] S. Verbrugge, D. Colle, M. Pickavet, P. Demeester, S. Pasqualini, A. Iselt, A. Kirstädter, R. Hülsermann, F.-J. Westphal, and M. Jäger, "Methodology and input availability parameters for calculating OpEx and CapEx costs for realistic network scenarios," *Journal of Optical Networking*, vol. 5, no. 6, p. 509, 2006. [Online]. Available: <https://www.osapublishing.org/jon/abstract.cfm?uri=jon-5-6-509>
- [136] B. Todd and J. Doucette, "A Novel Long Term Telecommunication Network Planning Framework," *Journal of Network and Systems Management*, vol. 25, no. 1, pp. 47–82, jan 2017. [Online]. Available: <http://link.springer.com/10.1007/s10922-016-9382-z>